




---

 AHELP for CIAO 3.4

## set\_state

Context: [varmm](#)

*Jump to:* [Description](#) [Example](#) [Bugs](#) [See Also](#)

---

### Synopsis

Set multiple fields of a state (configuration) variable

### Syntax

```
set_state( String_Type, Struct_Type )
```

### Description

The `set_state()` function provides an easy way to reset all or part of a CIAO state (also called configuration) variable in one call. The first parameter must be the name of the state variable, and accepts the same values as `set_state_defaults()`.

Although the second parameter can be a copy of a state object (as shown in the example below) – it need not be. Only those fields in the input structure that are present in the state object are used, other fields are silently ignored. If the field names match but the datatypes do not, then a warning message is issued and the field is ignored. Therefore

```
chips> set_state_defaults("varmm")
chips> copy = @varmm
chips> varmm.caseinsen = 1
chips> set_state( "varmm", copy );
chips> print(varmm)
readheader      = 1
rawkeys         = 1
verbosity       = 1
caseinsen       = 0
```

works, but the following produces an error

```
chips> temp = struct { caseinsen }
chips> temp.caseinsen = "a string"
chips> set_state( "varmm", temp );
field ignored: caseinsen          reason: type mismatch
```

## Example

We define a function that will plot a line using a fixed set of plot options – namely in red, using the "dot dash" linestyle – but that ensures the ChIPS state object is restored to its initial values when the routine returns.

```
define draw_line(x1,y1,x2,y2) {
  % copy the current ChIPS state object
  variable orig_state = @chips;

  % set the desired line options
  chips.linecolor = _chips->red;
  chips.linestyle = _chips->dotdash;

  % draw the line
  () = chips_line( x1, y1, x2, y2 );

  % restore the original state object
  set_state( "chips", orig_state );

  % return
  return;
}
```

Assuming this function is stored in a file called `lplot.sl`, the following shows how the ChIPS state object is unaffected by the call to `draw_line()`, since the third line is drawn with the same attributes (green, solid line) as the first one.

```
chips> () = evalfile("lplot.sl");
chips> clear
chips> limits 0 10 0 10
chips> chips.linecolor = _chips->green
chips> line 1 1 10 10
chips> draw_line(1,10,10,1)
chips> line 0 0 8 2
```

## Bugs

See the [bugs page for the Varrrm library](#) on the CIAO website for an up-to-date listing of known bugs.

## See Also

*modules*

[varrrm](#)

*varrrm*

[set\\_state defaults](#)