

S-Lang Grace Module Reference

John E. Davis, jed@jedsoft.org

Aug 26, 2007

Contents

1	Overview	2
2	Tutorial	2
3	Grace Module Function Reference	3
3.1	grace_new	3
3.2	grace.plot	3
3.3	grace.focus	4
3.4	grace.oplot	5
3.5	grace.tick	5
3.6	grace.label	6
3.7	grace.save	7
3.8	grace.new_color	7
3.9	grace.get_colors	8
3.10	grace.multi	8
3.11	grace.viewport	9
3.12	grace.pagesize	9
3.13	grace.hplot	10
3.14	grace.ohplot	10
3.15	grace.xrange	11
3.16	grace.yrange	11
3.17	grace.world	11
3.18	grace.ylabel	12
3.19	grace.xlabel	12
3.20	grace.title	13
3.21	grace.subtitle	13
3.22	grace.redraw	13
3.23	grace.clear	14

3.24	grace.close	14
3.25	grace.kill	14
3.26	grace.logx	15
3.27	grace.logy	15
3.28	grace.linx	15
3.29	grace.liny	16
3.30	grace.legend	16
3.31	grace.xtick	17
3.32	grace.ytick	17
3.33	grace.new_color	18

1 Overview

Grace <http://plasma-gate.weizmann.ac.il/Grace/> (formerly Xmgr) is a popular plotting program for Unix that features a GUI for WYSIWYG generation of publication quality 2-D graphs. The `grace`-module provides a high-level interface between the *S-lang* <http://www.jedsoft.org/slang/> interpreter and the *Grace* plotting application, permitting the creation of multiple *Grace* instances and allowing an arbitrary number of *Grace* windows to be open.

The module was developed and tested with Grace-5.1.20, using Debian “Etch” as the underlying OS. As the module makes extensive use of qualifiers, it will not work for versions of *S-lang* prior to 2.1. It is also unlikely to work with newer development versions of Grace (`grace-6`) because the mechanism used for communicating with Grace-5 appears to be unsupported in Grace-6. The module interacts with *Grace* via a one-way pipe that feeds it commands using the approach recommended in the *Grace User’s Guide* <http://plasma-gate.weizmann.ac.il/Grace/doc/UsersGuide.html>. As this pipe is one-way, it is impossible for the module to query *Grace* about its plotting state. For this reason, the module tries to maintain enough state to avoid potential conflicts with Grace. For example, the module will not instruct Grace to switch to a log scale without first making sure that no negative values have been plotted. If negative values were plotted, the module will first change the world-coordinate system to values that are appropriate for log plots.

The development for this release has been focused towards the types of 2-D graphs that are common in the physical sciences—specifically, Cartesian (X,Y) plots with various combinations of error-bars. While the module can be used to produce pie and bar charts, no work has been done to facilitate the construction of such graphs. A future version of the module will address these shortcomings.

2 Tutorial

A tutorial is available online from

<http://www.jedsoft.org/slang/modules/grace/tutorial.html>. It will not be repeated here.

3 Grace Module Function Reference

3.1 `grace_new`

Synopsis

Creates a new grace plotting instance

Usage

```
Struct_Type grace_new ( [args] )
```

Description

This function creates a new instance of a grace plotting object. When called with no arguments, it starts up `xmgrace` using the `-nosafe` command-line argument. If a different set of arguments are desired, then those may be specified as an array of strings. For example, to start Grace via `xmgrace -fixed -nosafe`, use

```
g = grace_new (["xmgrace", "-fixed", "-nosafe"]);
```

Note that the name of the application to run (`xmgrace` in this case) must be the first element in the array. If the `DISPLAY` environment variable is not set, the `gracebat` application will be started instead.

See Also

[3.2](#) (`grace.plot`), [3.24](#) (`grace.close`), [3.7](#) (`grace.save`)

3.2 `grace.plot`

Synopsis

Plot points or lines

Usage

```
.plot (x [,y [,dy]] [;qualifiers])
```

Description

The `plot` method erases the currently active plot and then plots the specified points. If called with a single argument, that argument will be used as the y points, and `[1,2,...]` will be used as the x points. If called with 3 arguments, the third argument will be used as an error bar.

Qualifiers

<code>logx [=0 1]</code>	Turn on/off log scaling for the x axis
<code>logy [=0 1]</code>	Turn on/off log scaling for the y axis
<code>line=int</code>	Linestyle (0 = no line)
<code>color=int str</code>	Line color
<code>width=float</code>	Line width
<code>fill=int</code>	
<code>fillcolor=int</code>	Color for the fill region
<code>fillpat=int</code>	Pattern index for filling
<code>sym=int</code>	Plot symbol (0 = none)
<code>symcolor=int str</code>	Symbol color

<code>symsize=float</code>	Symbol size (float)
<code>symfillcolor=int str</code>	Color used to fill symbols
<code>symfill=int</code>	
<code>grid[=0 1 2 3]</code>	Draw grid: 0:none, 1:major, 2:minor, 3:major+minor
<code>dropline</code>	Draw a dropline for each point
<code>graphtype=str</code>	One of "xy", "ydx", "xydy", "ydxxy", etc..

The following qualifiers control various attributes of the errorbars:

<code>errorbar_place=str</code>
<code>errorbar_color=int str</code>
<code>errorbar_pattern=int</code>
<code>errorbar_linestyle=int</code>
<code>errorbar_linewidth=float</code>
<code>errorbar_riser_linewidth=float</code>
<code>errorbar_riser_linestyle=int</code>
<code>errorbar_riser_clip_length=float</code>

Note: Using a negative value for `sym` is equivalent to setting `line=0` and `sym=abs(sym)`.

Example

```
x = [-10:10:#50];
y = sin(x);
g = grace_new ();
g.plot (x, y ; sym=-4, symfill=2);
```

Notes

Use the `.get_colors` method to see what colors are available.

See Also

[3.1](#) (`grace_new`), [3.4](#) (`grace.oplot`), [3.13](#) (`grace.hplot`), [3.10](#) (`grace.multi`), [3.5](#) (`grace.tick`)

3.3 `grace.focus`

Synopsis

Set the focus to a specified plot

Usage

```
grace.focus ( [ [nth] | [nrow,ncol] ] )}
```

Description

This method is used to change the focus of the plotting commands to a specified graph in a multi-plot situation. When called without arguments, the focus will be changed to the next graph. When called with the single argument `nth`, the focus will be set to the `nth` plot (`nth=1,2,..`). When called with 2 arguments, the first argument specifies the row and the second argument specifies the column (`row,col = 1,2,..`).

Example

```
g = grace_new ();
g.multi (2, 1; sizes=[3,1]);
g.focus (1,1);
g.plot (x,y);
g.focus (2,1);
g.plot (x,sin(x*y));
```

See Also

[3.10](#) (`grace.multi`)

3.4 `grace.oplot`

Synopsis

Overplot points or lines

Usage

```
.oplot (x [,y [,dy]] [;qualifiers])
```

Description

The `oplot` method may be used to add additional points or lines to a plot. It behaves like `plot` except that it does not erase the current plot. See the `plot` method documentation for more details.

See Also

[3.2](#) (`grace.plot`), [3.23](#) (`grace.clear`)

3.5 `grace.tick`

Synopsis

Control the axis tick marks

Usage

```
.tick ([enable_major [,enable_minor]] [;qualifiers])
```

Description

This method may be used to control various aspects of the ticks marks. If called with one argument, the major ticks may be turned on or off according to whether the argument is non-zero or not. A non-zero value turns on the generation of major ticks, and a value of zero turns off their generation. The optional second parameter controls the generation of minor ticks in a similar manner. Most other attributes are controlled by qualifiers.

Qualifiers

<code>offsetx=int</code>	offset the axis in the x direction by val
<code>offsety=int</code>	offset the axis in the y direction by val
<code>majorstyle=int</code>	linestyle for major ticks
<code>majorwidth=int</code>	line width for major ticks
<code>majorsize=int</code>	size of the major ticks
<code>majorcolor=val</code>	color of the major ticks

<code>minorstyle=int</code>	linestyle for minor ticks
<code>minorwidth=int</code>	line width for minor ticks
<code>minorsize=int</code>	size of the minor ticks
<code>minorcolor=val</code>	color of the minor ticks
<code>majorgrid=0 1</code>	enable(1) or disable the major grid
<code>minorgrid=0 1</code>	enable(1) or disable the minor grid

Example

```
g = grace_new ();
g.tick(;minorstyle=2);
g.plot(x,y;grid=3);
```

Notes

The tick method operates on both the x and y axes. Use the `xtick` and `ytick` methods for control over the corresponding axes.

The linestyle, width, and color of the major and minor grid lines are controlled by the corresponding tick attributes.

See Also

[3.31](#) (`grace.xtick`), [3.32](#) (`grace.ytick`), [3.2](#) (`grace.plot`)

3.6 `grace.label`**Synopsis**

Add a text label to a graph

Usage

```
.label (string, x, y [;qualifiers])
```

Description

This method may be used to add a text label to the current graph at the specified world coordinate.

Qualifiers

<code>viewport</code>	Use viewport coordinates instead of world coordinates
<code>world</code>	Use world coordinates (default)
<code>rot=angle</code>	Rotate the text by the specified angle
<code>font=int</code>	Use the specified font
<code>just=int</code>	justification: left=0, right=1, 2=center
<code>size=float</code>	Character size to use (default=1)

Notes

It is not a good idea to mix calls to this method with the creation of new text strings via the GUI. As Grace does not permit an external program to delete individual strings, the grace module keeps an internal table of string ids that have been used. New strings created via the GUI invalidate the table, and there is no way for the module to know that this has happened.

See Also

[3.19](#) (`grace.xlabel`), [3.20](#) (`grace.title`), [3.18](#) (`grace.ylabel`)

3.7 `g.save`

Synopsis

Save the plot to a file

Usage

```
.save(filename [;device])
```

Description

This method saves the plot to the specified filename. Normally the file format is determined from the filename extension, e.g., `foo.ps` indicates that `postscript` format is to be used. This `device` qualifier may be used to explicitly specify the format.

Example

```
g.save ("foo.eps");           % Encapsulated Postscript
g.save ("foo.agr");           % Grace format
g.save ("foo.out"; device=png);
```

See Also

[3.24](#) (`g.close`)

3.8 `g.new_color`

Synopsis

Add a color to the colormap

Usage

```
.new_color (name, rgb)
```

Description

The `new_color` method may be used to define a new color name, or redefine an existing color. The `rgb` parameter must be an integer that encodes the RGB value to be associated with the name. The `id` qualifier may be used to map the color to a specific color id value.

Example

```
g.new_color ("royalblue", 0x4169E1);
g.new_color ("royalblue", (65 shl 16)|(105 shl 8)|225);
```

The latter example shows how to construct the rgb-encoded integer from decimal rgb values.

See Also

[3.2](#) (`g.plot`), [3.9](#) (`g.get_colors`)

3.9 `grace.get_colors`

Synopsis

Get a list of the color names

Usage

```
String_Type[] = .get_colors()
```

Description

This method returns an array of the color names that have been defined.

Example

```
slsh> print (g.get_colors());
```

See Also

[3.33](#) (`grace.new_color`), [3.2](#) (`grace.plot`)

3.10 `grace.multi`

Synopsis

Arrange graphs into a rectangular grid

Usage

```
.multi(nrows,ncols [;qualifiers])
```

Description

This method may be used to layout graphs onto a rectangular grid consisting of `nrows` and `ncols`. Qualifiers may be used to specify the horizontal and vertical separations of the graphs.

Qualifiers

<code>vgap=value</code>	Controls the vertical spacing between graph windows.
<code>hgap=value</code>	Controls the horizontal spacing between graph windows.
<code>offset=value</code>	Sets the distance from the page edges.

The values for the `vgap` and `hgap` qualifiers appear to be fractions of the graph viewport size. The default values are `vgap=0.3` and `hgap=0.3`. The value of the `offset` qualifier appears to a fraction of the page size; the default value is 0.15.

See Also

[3.11](#) (`grace.viewport`)

3.11 `grace.viewport`

Synopsis

Set the viewport for the current graph

Usage

```
.viewport(xmin,xmax,ymin,ymax)
```

Description

This method may be used to set the viewport for the current graph. Each of the values must be expressed as a fraction of the longer page-size value. For example, if the page size is 8.5 by 11 inches, in portrait mode, then the X viewport coordinate runs from 0 to $8.5/11=0.773$, and the Y coordinate runs from 0 to 1. The rationale behind this stems from the desire to keep the physical size of the viewport an invariant with respect to landscape or portrait mode.

See Also

[3.12](#) (`grace.pagesize`)

3.12 `grace.pagesize`

Synopsis

Set the page size

Usage

```
.pagesize(dx, dy | "letter" | "a4" [;qualifiers])
```

Description

This method may be used to set the size of the plotting page. The value `dx` and `dy` are interpreted as $1/72$ of an inch by some plot devices, and as pixels by others. Alternatively, a single string such "a4" or "letter" may be used to select the corresponding page size.

After calling this method it is a good idea to re-adjust the viewport. An easy way to do this is to use the `.multi` method.

Qualifiers

<code>units="cm" "in"</code>	Interpret dx and dy in the corresponding units
<code>landscape</code>	Use landscape mode
<code>portrait</code>	Use portrait mode

Example

```
g.pagesize (14, 12 ; units="cm");  
g.multi(1,1);
```

See Also

[3.11](#) (`grace.viewport`), [3.10](#) (`grace.multi`)

3.13 `grace.hplot`

Synopsis

Plot a histogram

Usage

```
.hplot(x, y [,dy1 [,dy2]] [;qualifiers])
```

Description

This method is typically used to to plot binned data. In this context, `x` is an array that represents the lower bin edges and `y` is an array that gives the value of each bin. If the optional third or fourth arguments are present, the corresponding error bars (symmetric or asymmetric) will also be plotted.

Qualifiers

The `hplot` method accepts the same set of qualifiers as the the `plot` method. However, the `sym` qualifier is ignored unless error-bars are drawn, in which case the symbol is associated with the error bar.

Example

```
xbins = [1:1024:8];  
h = hist1d (xbins, pnts);  
g.plot (xbins, h, sqrt(h) ; color="red", sym=2);
```

Notes

The array `x` that specifies the lower bin edges must be in an increasing order such that $x[j+1] > x[j]$.

See Also

[3.2](#) (`grace.plot`), [3.14](#) (`grace.ohplot`)

3.14 `grace.ohplot`

Synopsis

Overplot a histogram

Usage

```
.ohplot()
```

Description

This method may be used to add a histogram plot to a graph. See the `hplot` method for additional information.

See Also

[3.13](#) (`grace.hplot`), [3.4](#) (`grace.oplot`)

3.15 `grace.xrangle`

Synopsis

Set the world coordinates for the X axis

Usage

```
.xrangle(xmin, xmax)
```

Description

This method may be used to assign world coordinate ranges for the X axis. If `xmin` or `xmax` is `NULL`, the corresponding limit will be computed from the data.

Example

```
xrangle (min(x), max(x));  
xrangle (,max(x));      % Set just the upper limit  
xrangle (,);           % Turn on autoscaling for this axis
```

See Also

[3.16](#) (`grace.yrange`), [3.17](#) (`grace.world`)

3.16 `grace.yrange`

Synopsis

Set the world coordinates for the Y axis

Usage

```
.yrange()
```

Description

This method may be used to assign world coordinate ranges for the Y axis. If `ymin` or `ymax` is `NULL`, the corresponding limit will be computed from the data.

Example

```
xrangle (min(y), max(y));  
xrangle (,max(y));      % Set just the upper limit  
xrangle (,);           % Turn on autoscaling for this axis
```

See Also

[3.15](#) (`grace.xrangle`), [3.17](#) (`grace.world`)

3.17 `grace.world`

Synopsis

Set the world coordinates

Usage

```
.world( [ xmin,xmax,ymin,ymax ])
```

Description

This method may be used to assign world coordinate ranges for the X and Y axes of the current graph. If any of these values is NULL, the corresponding limit will be computed from the data (autoscaling). When called without arguments, all limits will be computed from the data.

See Also

[3.15](#) (`grace.xrangle`), [3.16](#) (`grace.yrange`), [3.11](#) (`grace.viewport`)

3.18 `grace.ylabel`**Synopsis**

Add the Y-axis label

Usage

```
.ylabel(string)
```

Description

This method may be used to add a label to the Y axis.

Qualifiers

<code>color=int str</code>	Label color
<code>size=float</code>	Label Size
<code>font=int</code>	Font id

See Also

[3.19](#) (`grace.xlabel`), [3.20](#) (`grace.title`), [3.6](#) (`grace.label`)

3.19 `grace.xlabel`**Synopsis**

Add the X-axis label

Usage

```
.xlabel()
```

Description

This method may be used to add a label to the X axis.

Qualifiers

<code>color=int str</code>	Label color
<code>size=float</code>	Label Size
<code>font=int</code>	Font id

See Also

[3.18](#) (`grace.ylabel`), [3.20](#) (`grace.title`), [3.6](#) (`grace.label`)

3.20 `grace.title`

Synopsis

Add the title to the graph

Usage

```
.title(string)
```

Description

This method may be used to add a title to the top of the current graph.

Qualifiers

<code>color=int str</code>	Label color
<code>size=float</code>	Label Size
<code>font=int</code>	Font id

See Also

[3.19](#) (`grace.xlabel`), [3.18](#) (`grace.ylabel`), [3.21](#) (`grace.subtitle`), [3.6](#) (`grace.label`)

3.21 `grace.subtitle`

Synopsis

Add the subtitle to the graph

Usage

```
.subtitle(string)
```

Description

This method may be used to add a subtitle to the top of the current graph under the main graph title.

Qualifiers

<code>color=int str</code>	Label color
<code>size=float</code>	Label Size
<code>font=int</code>	Font id

See Also

[3.19](#) (`grace.xlabel`), [3.18](#) (`grace.ylabel`), [3.20](#) (`grace.title`), [3.6](#) (`grace.label`)

3.22 `grace.redraw`

Synopsis

Redraw the Grace display

Usage

```
.redraw()
```

Description

This method is sometimes necessary to tell Grace to redraw or refresh display.

See Also

[3.2](#) (`grace.plot`)

3.23 `grace.clear`**Synopsis**

Clear the current graph

Usage

```
.clear()
```

Description

This function may be used to clear the current graph. Is it not normally needed since the `plot` and `hplot` methods automatically do this.

See Also

[3.25](#) (`grace.kill`), [3.24](#) (`grace.close`), [3.2](#) (`grace.plot`)

3.24 `grace.close`**Synopsis**

Close the grace session

Usage

```
.close()
```

Description

This method is used to shutdown the Grace GUI. Unless this function is called, the GUI will remain active even after the *S-lang* application has exited.

See Also

[3.1](#) (`grace_new`), [3.7](#) (`grace.save`)

3.25 `grace.kill`**Synopsis**

Delete the current graph

Usage

```
.kill()
```

Description

This method may be used to delete the current graph. It can be useful in a multi-plot situation when some graphs are not needed.

See Also

[3.23](#) (`grace.clear`), [3.24](#) (`grace.close`)

3.26 `grace.logx`**Synopsis**

Change the X axis to a log scale

Usage

```
.logx()
```

Description

This method causes the X axis to use log scaling. If the currently focused graph has a world coordinate system that conflicts with log scaling, the world coordinates will be changed accordingly (restricted to positive values).

Notes

Normally log scaling is specified when the plot is made via the `plot` method's `logx` qualifier.

3.27 `grace.logy`**Synopsis**

Change the Y axis to a log scale

Usage

```
.logy()
```

Description

This method causes the Y axis to use log scaling. If the currently focused graph has a world coordinate system that conflicts with log scaling, the world coordinates will be changed accordingly (restricted to positive values).

Notes

Normally log scaling is specified when the plot is made via the `plot` method's `logy` qualifier.

See Also

[3.26](#) (`grace.logx`), [3.28](#) (`grace.linx`), [3.29](#) (`grace.liny`)

3.28 `grace.linx`**Synopsis**

Change the X axis to a linear scale

Usage

```
.linx()
```

Description

This method may be used to cause the X axis of the current graph to use linear scaling.

See Also

[3.29](#) (`grace.liny`), [3.26](#) (`grace.logx`), [3.27](#) (`grace.logy`)

3.29 `grace.liny`**Synopsis**

Change the Y axis to a linear scale

Usage

```
.liny()
```

Description

This method may be used to cause the Y axis of the current graph to use linear scaling.

See Also

[3.28](#) (`grace.linx`), [3.26](#) (`grace.logx`), [3.27](#) (`grace.logy`)

3.30 `grace.legend`**Synopsis**

Add a legend to the current graph

Usage

```
.legend(x, y, names)
```

Description

This method may be used to add a legend to the graph at the coordinate (x,y). The legend is formed from the `names` array and representative lines or symbols used to draw the corresponding curves.

Qualifiers

These qualifiers control attribute of the objects in the legend box:

<code>font=int</code>	Font id
<code>size=float</code>	character size (default: 1)
<code>color=int str</code>	color
<code>vgap=int</code>	
<code>hgap=int</code>	
<code>length=int</code>	

The following qualifiers control the legend box itself:

<code>loctype=str</code>	"world" or "view" (default: "world")
<code>box=0 1</code>	Turn the box off(0) or on(1)
<code>box_line=int</code>	Line style
<code>box_color=int str</code>	Line color
<code>box_linewidth=float</code>	

```
box_pattern=int  
box_fillcolor=int|str  
box_fillpattern=int
```

Example

```
g.plot (x,f(x));  
g.oplot (x,g(x));  
g.legend (2.1, 3.9, ["f(x)", "g(x)"]; box=0);
```

See Also

[3.6](#) (`grace.label`), [3.20](#) (`grace.title`), [3.2](#) (`grace.plot`)

3.31 `grace.xtick`

Synopsis

Adjust the X axis ticks

Usage

```
.xtick()
```

Description

This method may be used to control certain attributes of the X axis tick marks. See the documentation for the `tick` method for more information.

See Also

[3.5](#) (`grace.tick`), [3.32](#) (`grace.ytick`)

3.32 `grace.ytick`

Synopsis

Adjust the Y axis ticks

Usage

```
.ytick()
```

Description

This method may be used to control certain attributes of the Y axis tick marks. See the documentation for the `tick` method for more information.

See Also

[3.5](#) (`grace.tick`), [3.32](#) (`grace.ytick`)

3.33 `grace.new_color`

Synopsis

Define a new color name

Usage

```
.new_color(name, RGB)
```

Description

This method may be used to add a new named color or redefine a named color. The first parameter is a string that specifies the name of the color, and the second is a 3 element integer array whose successive elements specify the Red, Green, and Blue content of the color. 0-255. for the color.

Qualifiers

id=int Map the named color to the specified color id or index

Example

```
g.new_color("skyblue3", [108, 166, 205]);  
g.plot (x,y; color="skyblue3");
```

See Also

[3.9](#) (`grace.get_colors`), [3.2](#) (`grace.plot`)