# fplot

1.0.0

Generated by Doxygen 1.8.11

# Contents

**1 Modules Index** **2**

1.1 Modules List . . . 2

**2 Data Type Index** **2**

2.1 Class Hierarchy . . . 2

**3 Data Type Index** **3**

3.1 Data Types List . . . 3

**4 Module Documentation** **5**

4.1 fplot_core Module Reference . . . 5

4.1.1 Detailed Description . . . 15

4.1.2 Function/Subroutine Documentation . . . 16

4.2 fplot_errors Module Reference . . . 65

4.2.1 Detailed Description . . . 65

4.3 fplot_list Module Reference . . . 65

4.3.1 Detailed Description . . . 66

4.3.2 Function/Subroutine Documentation . . . 66

**5 Data Type Documentation** **70**

5.1 fplot_core::cm_get_string_result Interface Reference . . . 70

5.1.1 Detailed Description . . . 70

5.2 fplot_core::color Type Reference . . . 71

5.2.1 Detailed Description . . . 71

5.3 fplot_core::colormap Type Reference . . . 71

5.4 fplot_list::container Type Reference . . . 71

5.4.1 Detailed Description . . . 72

5.5 fplot_core::cool_colormap Type Reference . . . 72

5.5.1 Detailed Description . . . 72

5.6 fplot_core::get_string_result Interface Reference . . . 72

5.6.1 Detailed Description . . . 72

5.7 fplot_core::hot_colormap Type Reference . . . 73
5.7.1 Detailed Description . . . 73
5.8 fplot_core::legend Type Reference . . . 73
5.8.1 Detailed Description . . . 74
5.9 fplot_list::list Type Reference . . . 75
5.9.1 Detailed Description . . . 75
5.10 fplot_core::pa_get_string_result Interface Reference . . . 75
5.10.1 Detailed Description . . . 76
5.11 fplot_core::pd_get_string_result Interface Reference . . . 77
5.11.1 Detailed Description . . . 77
5.12 fplot_core::plot Type Reference . . . 77
5.12.1 Detailed Description . . . 79
5.13 fplot_core::plot_2d Type Reference . . . 79
5.13.1 Detailed Description . . . 80
5.14 fplot_core::plot_3d Type Reference . . . 80
5.14.1 Detailed Description . . . 81
5.15 fplot_core::plot_axis Type Reference . . . 81
5.15.1 Detailed Description . . . 83
5.16 fplot_core::plot_data Type Reference . . . 83
5.16.1 Detailed Description . . . 83
5.17 fplot_core::plot_data_2d Type Reference . . . 83
5.17.1 Detailed Description . . . 84
5.18 fplot_core::plot_data_3d Type Reference . . . 85
5.18.1 Detailed Description . . . 85
5.19 fplot_core::plot_object Type Reference . . . 86
5.19.1 Detailed Description . . . 86
5.20 fplot_core::png_terminal Type Reference . . . 86
5.20.1 Detailed Description . . . 87
5.21 fplot_core::qt_terminal Type Reference . . . 87
5.21.1 Detailed Description . . . 87

5.22 fplot_core::rainbow_colormap Type Reference . . . 87
5.22.1 Detailed Description . . . 88
5.23 fplot_core::scatter_plot_data Type Reference . . . 88
5.23.1 Detailed Description . . . 90
5.24 fplot_core::spd_get_int_value Interface Reference . . . 90
5.24.1 Detailed Description . . . 90
5.25 fplot_core::spd_get_string_result Interface Reference . . . 90
5.25.1 Detailed Description . . . 90
5.26 fplot_core::spd_get_value Interface Reference . . . 91
5.26.1 Detailed Description . . . 91
5.27 fplot_core::spd_set_value Interface Reference . . . 91
5.27.1 Detailed Description . . . 92
5.28 fplot_core::surface_plot Type Reference . . . 92
5.28.1 Detailed Description . . . 93
5.29 fplot_core::surface_plot_data Type Reference . . . 93
5.29.1 Detailed Description . . . 94
5.30 fplot_core::term_get_string_result Interface Reference . . . 94
5.30.1 Detailed Description . . . 94
5.31 fplot_core::terminal Type Reference . . . 95
5.31.1 Detailed Description . . . 96
5.32 fplot_core::windows_terminal Type Reference . . . 96
5.32.1 Detailed Description . . . 97
5.33 fplot_core::wxt_terminal Type Reference . . . 97
5.33.1 Detailed Description . . . 97
5.34 fplot_core::x_axis Type Reference . . . 97
5.34.1 Detailed Description . . . 98
5.35 fplot_core::y2_axis Type Reference . . . 98
5.35.1 Detailed Description . . . 98
5.36 fplot_core::y_axis Type Reference . . . 99
5.36.1 Detailed Description . . . 99
5.37 fplot_core::z_axis Type Reference . . . 99
5.37.1 Detailed Description . . . 99

Index 101

# 1 Modules Index

## 1.1 Modules List

Here is a list of all documented modules with brief descriptions:

fplot_core
fplot_core 5

fplot_errors
plot_errors 65

fplot_list
fplot_list 65

# 2 Data Type Index

## 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

fplot_core::cm_get_string_result 70

fplot_core::color 71

fplot_list::container 71

fplot_core::get_string_result 72

fplot_list::list 75

fplot_core::pa_get_string_result 75

fplot_core::pd_get_string_result 77

fplot_core::plot_object 86

fplot_core::colormap 71

fplot_core::cool_colormap 72

fplot_core::hot_colormap 73

fplot_core::rainbow_colormap 87

fplot_core::legend 73

fplot_core::plot 77

fplot_core::plot_2d 79

fplot_core::plot_3d 80

fplot_core::surface_plot 92
fplot_core::plot_axis 81
fplot_core::x_axis 97
fplot_core::y2_axis 98
fplot_core::y_axis 99
fplot_core::z_axis 99
fplot_core::plot_data 83
fplot_core::scatter_plot_data 88
fplot_core::plot_data_2d 83
fplot_core::plot_data_3d 85
fplot_core::surface_plot_data 93
fplot_core::terminal 95
fplot_core::png_terminal 86
fplot_core::qt_terminal 87
fplot_core::windows_terminal 96
fplot_core::wxt_terminal 97
fplot_core::spd_get_int_value 90
fplot_core::spd_get_string_result 90
fplot_core::spd_get_value 91
fplot_core::spd_set_value 91
fplot_core::term_get_string_result 94

# 3 Data Type Index

## 3.1 Data Types List

Here are the data types with brief descriptions:

fplot_core::cm_get_string_result
Retrieves a string from a colormap 70

fplot_core::color
Describes an RGB color 71

fplot_core::colormap
A colormap object for a surface plot 71

fplot_list::container
A container type allowing storage of most any Fortran type 71

fplot_core::cool_colormap
Defines a colormap consisting of "cool" colors 72

fplot_core::get_string_result
Retrieves a string from a plot_object 72

fplot_core::hot_colormap
Defines a colormap consisting of "hot" colors 73

fplot_core::legend
Defines a legend object 73

fplot_list::list
A generic list container 75

fplot_core::pa_get_string_result
Retrieves a string from a plot_axis 75

fplot_core::pd_get_string_result
Retrieves a string from a plot_data object 77

fplot_core::plot
Defines the basic GNUPLOT plot 77

fplot_core::plot_2d
A plot object defining a 2D plot 79

fplot_core::plot_3d
A plot object defining a 3D plot 80

fplot_core::plot_axis
Describes a single plot axis 81

fplot_core::plot_data
Provides a container for plot data 83

fplot_core::plot_data_2d
Defines a two-dimensional plot data set 83

fplot_core::plot_data_3d
Defines a three-dimensional plot data set 85

fplot_core::plot_object
The base type for a GNUPLOT object 86

fplot_core::png_terminal
Defines a GNUPLOT PNG terminal object 86

fplot_core::qt_terminal
Defines a GNUPLOT QT terminal object 87

fplot_core::rainbow_colormap
Defines a rainbow colormap 87

fplot_core::scatter_plot_data
A plot_data object for describing scatter plot data sets 88

fplot_core::spd_get_int_value
Retrieves an integer value from a scatter_plot_data object 90

fplot_core::spd_get_string_result
Retrieves a string from a scatter_plot_data object 90

**fplot_core::spd_get_value**
Retrieves a numeric value from a scatter_plot_data object 91

**fplot_core::spd_set_value**
Sets a numeric value into a scatter_plot_data object 91

**fplot_core::surface_plot** 92

**fplot_core::surface_plot_data**
Provides a three-dimensional surface plot data set 93

**fplot_core::term_get_string_result**
Retrieves a string from a terminal 94

**fplot_core::terminal**
Defines a GNUPLOT terminal object 95

**fplot_core::windows_terminal**
Defines a GNUPLOT Win32 terminal object 96

**fplot_core::wxt_terminal**
Defines a GNUPLOT WXT terminal object 97

**fplot_core::x_axis**
An x-axis object 97

**fplot_core::y2_axis**
A secondary y-axis object 98

**fplot_core::y_axis**
A y-axis object 99

**fplot_core::z_axis**
A z-axis object 99

# 4 Module Documentation

## 4.1 fplot_core Module Reference

**fplot_core**

### Data Types

- interface cm_get_string_result
  *Retrieves a string from a colormap.*
- type color
  *Describes an RGB color.*
- type colormap
  *A colormap object for a surface plot.*
- type cool_colormap
  *Defines a colormap consisting of "cool" colors.*
- interface get_string_result
  *Retrieves a string from a plot_object.*
- type hot_colormap

*Defines a colormap consisting of "hot" colors.*

- type legend

  *Defines a legend object.*
- interface pa_get_string_result

  *Retrieves a string from a plot_axis.*
- interface pd_get_string_result

  *Retrieves a string from a plot_data object.*
- type plot

  *Defines the basic GNUPLOT plot.*
- type plot_2d

  *A plot object defining a 2D plot.*
- type plot_3d

  *A plot object defining a 3D plot.*
- type plot_axis

  *Describes a single plot axis.*
- type plot_data

  *Provides a container for plot data.*
- type plot_data_2d

  *Defines a two-dimensional plot data set.*
- type plot_data_3d

  *Defines a three-dimensional plot data set.*
- type plot_object

  *The base type for a GNUPLOT object.*
- type png_terminal

  *Defines a GNUPLOT PNG terminal object.*
- type qt_terminal

  *Defines a GNUPLOT QT terminal object.*
- type rainbow_colormap

  *Defines a rainbow colormap.*
- type scatter_plot_data

  *A plot_data object for describing scatter plot data sets.*
- interface spd_get_int_value

  *Retrieves an integer value from a scatter_plot_data object.*
- interface spd_get_string_result

  *Retrieves a string from a scatter_plot_data object.*
- interface spd_get_value

  *Retrieves a numeric value from a scatter_plot_data object.*
- interface spd_set_value

  *Sets a numeric value into a scatter_plot_data object.*
- type surface_plot
- type surface_plot_data

  *Provides a three-dimensional surface plot data set.*
- interface term_get_string_result

  *Retrieves a string from a terminal.*
- type terminal

  *Defines a GNUPLOT terminal object.*
- type windows_terminal

  *Defines a GNUPLOT Win32 terminal object.*
- type wxt_terminal

  *Defines a GNUPLOT WXT terminal object.*
- type x_axis

*An x-axis object.*

- type y2_axis

  *A secondary y-axis object.*
- type y_axis

  *A y-axis object.*
- type z_axis

  *A z-axis object.*

#### Functions/Subroutines

- pure character(6) function clr_to_hex_string (this)

  *Returns the color in hexadecimal format.*
- subroutine clr_copy_from (this, clr)

  *Copies another color to this color.*
- pure integer function term_get_window_width (this)

  *Gets the width of the plot window.*
- subroutine term_set_window_width (this, x)

  *Sets the width of the plot window.*
- pure integer function term_get_window_height (this)

  *Gets the height of the plot window.*
- subroutine term_set_window_height (this, x)

  *Sets the height of the plot window.*
- pure integer(int32) function term_get_plot_window_number (this)

  *Gets the targeted plot window number.*
- subroutine term_set_plot_window_number (this, x)

  *Sets the targeted plot window number.*
- pure character(len=:) function, allocatable term_get_title (this)

  *Gets the plot window's title.*
- subroutine term_set_title (this, txt)

  *Sets the plot window's title.*
- pure character(len=:) function, allocatable term_get_font_name (this)

  *Gets the name of the font used for text displayed by the graph.*
- subroutine term_set_font_name (this, name)

  *Sets the name of the font used for text displayed by the graph.*
- pure integer function term_get_font_size (this)

  *Gets the size of the font used by the graph.*
- subroutine term_set_font_size (this, sz)

  *Sets the size of the font used by the graph.*
- character(len=:) function, allocatable term_get_command_string (this)

  *Returns the appropriate GNUPLOT command string to establish appropriate parameters.*
- pure character(len=:) function, allocatable wt_get_term_string (this)

  *Retrieves a GNUPLOT terminal identifier string.*
- pure character(len=:) function, allocatable qt_get_term_string (this)

  *Retrieves a GNUPLOT terminal identifier string.*
- pure character(len=:) function, allocatable wxt_get_term_string (this)

  *Retrieves a GNUPLOT terminal identifier string.*
- pure character(len=:) function, allocatable png_get_term_string (this)

  *Retrieves a GNUPLOT terminal identifier string.*
- pure character(len=:) function, allocatable png_get_filename (this)

  *Gets the filename for the output PNG file.*

- subroutine png_set_filename (this, txt)

  *Sets the filename for the output PNG file.*
- character(len=:) function, allocatable png_get_command_string (this)

  *Returns the appropriate GNUPLOT command string to establish appropriate parameters.*
- pure character(len=:) function, allocatable pd_get_name (this)

  *Gets the name to associate with this data set.*
- subroutine pd_set_name (this, txt)

  *Sets the name to associate with this data set.*
- pure character(len=:) function, allocatable pa_get_title (this)

  *Gets the axis' title.*
- subroutine pa_set_title (this, txt)

  *Sets the axis' title.*
- pure logical function pa_has_title (this)

  *Gets a value determining if a title has been defined for the plot_axis object.*
- pure logical function pa_get_autoscale (this)

  *Gets a logical value determining if the axis should be automatically scaled to fit the data.*
- subroutine pa_set_autoscale (this, x)

  *Sets a logical value determining if the axis should be automatically scaled to fit the data.*
- pure real(real64) function, dimension(2) pa_get_axis_limits (this)

  *Gets the axis display limits, assuming autoscaling is not active for this axis.*
- subroutine pa_set_axis_limits (this, lower, upper)

  *Sets the axis display limits, assuming autoscaling is not active for this axis.*
- pure logical function pa_get_log_scale (this)

  *Gets a logical value defining if the axis should be log scaled.*
- subroutine pa_set_log_scale (this, x)

  *Sets a logical value defining if the axis should be log scaled.*
- character(len=:) function, allocatable pa_get_cmd_string (this)

  *Returns the appropriate GNUPLOT command string to define the plot_axis properties.*
- pure logical function pa_get_zero_axis (this)

  *Gets a value determining if the axis should be drawn through zero of opposing axes.*
- subroutine pa_set_zero_axis (this, x)

  *Sets a value determining if the axis should be drawn through zero of opposing axes.*
- pure real(real32) function pa_get_zero_axis_width (this)

  *Gets the width of the line used to represent the zero axis line, if active.*
- subroutine pa_set_zero_axis_width (this, x)

  *Gets the width of the line used to represent the zero axis line, if active.*
- pure logical function leg_get_inside (this)

  *Gets a value determining if the legend should be drawn inside the axes border (true), or outside the axes border (false).*
- subroutine leg_set_inside (this, x)

  *Sets a value determining if the legend should be drawn inside the axes border (true), or outside the axes border (false).*
- pure logical function leg_get_box (this)

  *Gets a value determining if the legend should have a border.*
- subroutine leg_set_box (this, x)

  *Sets a value determining if the legend should have a border.*
- pure character(len=:) function, allocatable leg_get_horz_pos (this)

  *Gets the horizontal position of the legend.*
- subroutine leg_set_horz_pos (this, x)

  *Sets the horizontal position of the legend.*
- pure character(len=:) function, allocatable leg_get_vert_pos (this)

*Gets the vertical position of the legend.*

- subroutine leg_set_vert_pos (this, x)

  *Sets the vertical position of the legend.*
- pure logical function leg_get_visible (this)

  *Gets a value determining if the legend is visible.*
- subroutine leg_set_visible (this, x)

  *Sets a value determining if the legend is visible.*
- character(len=:) function, allocatable leg_get_command_txt (this)

  *Gets the command string defining the legend properties.*
- subroutine plt_clean_up (this)

  *Cleans up resources held by the plot object.*
- subroutine plt_init (this, term, err)

  *Initializes the plot object.*
- pure character(len=:) function, allocatable plt_get_title (this)

  *Gets the plot's title.*
- subroutine plt_set_title (this, txt)

  *Sets the plot's title.*
- pure logical function plt_has_title (this)

  *Gets a value determining if a title has been defined for the plot object.*
- type(legend) function, pointer plt_get_legend (this)

  *Gets the plot's legend object.*
- pure integer(int32) function plt_get_count (this)

  *Gets the number of stored plot_data objects.*
- subroutine plt_push_data (this, x, err)

  *Pushes a plot_data object onto the stack.*
- subroutine plt_pop_data (this)

  *Pops the last plot_data object from the stack.*
- subroutine plt_clear_all (this)

  *Removes all plot_data objects from the plot.*
- class(plot_data) function, pointer plt_get (this, i)

  *Gets a pointer to the requested plot_data object.*
- subroutine plt_set (this, i, x)

  *Sets the requested plot_data object into the plot.*
- class(terminal) function, pointer plt_get_term (this)

  *Gets the GNUPLOT terminal object.*
- pure logical function plt_get_show_grid (this)

  *Gets a flag determining if the grid lines should be shown.*
- subroutine plt_set_show_grid (this, x)

  *Sets a flag determining if the grid lines should be shown.*
- subroutine plt_draw (this, persist, err)

  *Launches GNUPLOT and draws the plot per the current state of the command list.*
- subroutine plt_save (this, fname, err)

  *Saves a GNUPLOT command file.*
- character(len=:) function, allocatable plt_get_font (this)

  *Gets the name of the font used for plot text.*
- subroutine plt_set_font (this, x)

  *Sets the name of the font used for plot text.*
- integer(int32) function plt_get_font_size (this)

  *Gets the size of the font used by the plot.*
- subroutine plt_set_font_size (this, x)

  *Sets the size of the font used by the plot.*

- pure logical function plt_get_tics_in (this)

  *Gets a value determining if the axis tic marks should point inwards.*
- subroutine plt_set_tics_in (this, x)

  *Sets a value determining if the axis tic marks should point inwards.*
- pure logical function plt_get_draw_border (this)

  *Gets a value determining if the border should be drawn.*
- subroutine plt_set_draw_border (this, x)

  *Sets a value determining if the border should be drawn.*
- character(len=:) function, allocatable spd_get_cmd (this)

  *Gets the GNUPLOT command string to represent this scatter_plot_data object.*
- pure real(real32) function spd_get_line_width (this)

  *Gets the width of the line, in pixels.*
- subroutine spd_set_line_width (this, x)

  *Sets the width of the line, in pixels.*
- pure integer(int32) function spd_get_line_style (this)

  *Gets the line style.*
- subroutine spd_set_line_style (this, x)

  *Sets the line style.*
- pure type(color) function spd_get_line_color (this)

  *Gets the line color.*
- subroutine spd_set_line_color (this, x)

  *Sets the line color.*
- pure logical function spd_get_draw_line (this)

  *Gets a value determining if a line should be drawn.*
- subroutine spd_set_draw_line (this, x)

  *Sets a value determining if a line should be drawn.*
- pure logical function spd_get_draw_markers (this)

  *Gets a value determining if data point markers should be drawn.*
- subroutine spd_set_draw_markers (this, x)

  *Sets a value determining if data point markers should be drawn.*
- pure integer(int32) function spd_get_marker_style (this)

  *Gets the marker style.*
- subroutine spd_set_marker_style (this, x)

  *Sets the marker style.*
- pure real(real32) function spd_get_marker_scaling (this)

  *Gets the marker scaling.*
- subroutine spd_set_marker_scaling (this, x)

  *Sets the marker scaling.*
- pure integer(int32) function spd_get_marker_frequency (this)

  *Gets the marker frequency.*
- subroutine spd_set_marker_frequency (this, x)

  *Sets the marker frequency.*
- pure logical function spd_get_use_auto_colors (this)

  *Gets a value determining if GNUPLOT should automatically choose line colors.*
- subroutine spd_set_use_auto_colors (this, x)

  *Sets a value determining if GNUPLOT should automatically choose line colors.*
- subroutine p2d_clean_up (this)

  *Cleans up resources held by the plot_2d object.*
- subroutine p2d_init (this, term, err)

  *Initializes the plot_2d object.*
- character(len=:) function, allocatable p2d_get_cmd (this)

*Gets the GNUPLOT command string to represent this plot_2d object.*

- class(plot_axis) function, pointer p2d_get_x_axis (this)

  *Gets the x-axis object.*
- class(plot_axis) function, pointer p2d_get_y_axis (this)

  *Gets the y-axis object.*
- class(plot_axis) function, pointer p2d_get_y2_axis (this)

  *Gets the secondary y-axis object.*
- pure logical function p2d_get_use_y2 (this)

  *Gets a flag determining if the secondary y-axis should be displayed.*
- subroutine p2d_set_use_y2 (this, x)

  *Sets a flag determining if the secondary y-axis should be displayed.*
- character(len=:) function, allocatable xa_get_id (this)

  *Gets the axis identification string.*
- character(len=:) function, allocatable ya_get_id (this)

  *Gets the axis identification string.*
- character(len=:) function, allocatable y2a_get_id (this)

  *Gets the axis identification string.*
- character(len=:) function, allocatable za_get_id (this)

  *Gets the axis identification string.*
- character(len=:) function, allocatable pd2d_get_axes_cmd (this)

  *Gets the GNUPLOT command string defining which axes the data is to be plotted against.*
- character(len=:) function, allocatable pd2d_get_data_cmd (this)

  *Gets the GNUPLOT command string containing the actual data to plot.*
- pure integer(int32) function pd2d_get_data_count (this)

  *Gets the number of data points.*
- pure real(real64) function pd2d_get_x_data (this, index)

  *Gets the requested X data point.*
- subroutine pd2d_set_x_data (this, index, x)

  *Sets the requested X data point.*
- pure real(real64) function pd2d_get_y_data (this, index)

  *Gets the requested Y data point.*
- subroutine pd2d_set_y_data (this, index, x)

  *Sets the requested Y data point.*
- subroutine pd2d_set_data_1 (this, x, y, err)

  *Defines the data set.*
- pure logical function pd2d_get_draw_against_y2 (this)

  *Gets a value determining if the data should be plotted against the secondary y-axis.*
- subroutine pd2d_set_draw_against_y2 (this, x)

  *Sets a value determining if the data should be plotted against the secondary y-axis.*
- subroutine pd2d_set_data_2 (this, y, err)

  *Defines the data set.*
- pure integer(int32) function pd3d_get_data_count (this)

  *Gets the number of data points.*
- pure real(real64) function pd3d_get_x_data (this, index)

  *Gets the requested X data point.*
- subroutine pd3d_set_x_data (this, index, x)

  *Sets the requested X data point.*
- pure real(real64) function pd3d_get_y_data (this, index)

  *Gets the requested Y data point.*
- subroutine pd3d_set_y_data (this, index, x)

  *Sets the requested Y data point.*

- pure real(real64) function pd3d_get_z_data (this, index)

  *Gets the requested Z data point.*
- subroutine pd3d_set_z_data (this, index, x)

  *Sets the requested Z data point.*
- character(len=:) function, allocatable pd3d_get_axes_cmd (this)

  *Gets the GNUPLOT command string defining which axes the data is to be plotted against.*
- character(len=:) function, allocatable pd3d_get_data_cmd (this)

  *Gets the GNUPLOT command string containing the actual data to plot.*
- subroutine pd3d_set_data_1 (this, x, y, z, err)

  *Defines the data set.*
- subroutine p3d_clean_up (this)

  *Cleans up resources held by the plot_3d object.*
- subroutine p3d_init (this, term, err)

  *Initializes the plot_3d object.*
- character(len=:) function, allocatable p3d_get_cmd (this)

  *Gets the GNUPLOT command string to represent this plot_3d object.*
- class(plot_axis) function, pointer p3d_get_x_axis (this)

  *Gets the x-axis object.*
- class(plot_axis) function, pointer p3d_get_y_axis (this)

  *Gets the y-axis object.*
- class(plot_axis) function, pointer p3d_get_z_axis (this)

  *Gets the z-axis object.*
- pure real(real64) function p3d_get_elevation (this)

  *Gets the plot elevation angle.*
- subroutine p3d_set_elevation (this, x)

  *Sets the plot elevation angle.*
- pure real(real64) function p3d_get_azimuth (this)

  *Gets the plot azimuth angle.*
- subroutine p3d_set_azimuth (this, x)

  *Sets the plot azimuth angle.*
- pure logical function p3d_get_z_axis_intersect (this)

  *Gets a value determining if the z-axis should intersect the x-y plane.*
- subroutine p3d_set_z_axis_intersect (this, x)

  *Sets a value determining if the z-axis should intersect the x-y plane.*
- pure integer(int32) function surfd_get_size (this, dim)

  *Gets the size of the stored data set.*
- pure real(real64) function surfd_get_x (this, i, j)

  *Gets the requested X data point.*
- subroutine surfd_set_x (this, i, j, x)

  *Sets the requested X data point.*
- pure real(real64) function surfd_get_y (this, i, j)

  *Gets the requested Y data point.*
- subroutine surfd_set_y (this, i, j, x)

  *Sets the requested Y data point.*
- pure real(real64) function surfd_get_z (this, i, j)

  *Gets the requested Z data point.*
- subroutine surfd_set_z (this, i, j, x)

  *Sets the requested Z data point.*
- pure logical function surfd_get_wireframe (this)

  *Gets a value determining if a wireframe mesh should be displayed.*
- subroutine surfd_set_wireframe (this, x)

*Sets a value determining if a wireframe mesh should be displayed.*

- character(len=:) function, allocatable surfd_get_cmd (this)

  *Gets the GNUPLOT command string to represent this surface_plot_data object.*
- character(len=:) function, allocatable surfd_get_data_cmd (this)

  *Gets the GNUPLOT command string containing the actual data to plot.*
- subroutine surfd_set_data_1 (this, x, y, z, err)

  *Defines the data set.*
- subroutine surf_clean_up (this)

  *Cleans up resources held by the surface_plot object.*
- subroutine surf_init (this, term, err)

  *Initializes the surface_plot object.*
- pure logical function surf_get_show_hidden (this)

  *Gets a value indicating if hidden lines should be shown.*
- subroutine surf_set_show_hidden (this, x)

  *Sets a value indicating if hidden lines should be shown.*
- character(len=:) function, allocatable surf_get_cmd (this)

  *Gets the GNUPLOT command string to represent this surface_plot object.*
- class(colormap) function, pointer surf_get_colormap (this)

  *Gets a pointer to the colormap object.*
- subroutine surf_set_colormap (this, x, err)

  *Sets the colormap object.*
- pure logical function surf_get_smooth (this)

  *Gets a value determining if the plotted surfaces should be smoothed.*
- subroutine surf_set_smooth (this, x)

  *Sets a value determining if the plotted surfaces should be smoothed.*
- pure logical function surf_get_show_contours (this)

  *Gets a value determining if a contour plot should be drawn in conjunction with the surface plot.*
- subroutine surf_set_show_contours (this, x)

  *Sets a value determining if a contour plot should be drawn in conjunction with the surface plot.*
- pure logical function surf_get_show_colorbar (this)

  *Gets a value determining if the colorbar should be shown.*
- subroutine surf_set_show_colorbar (this, x)

  *Sets a value determining if the colorbar should be shown.*
- character(len=:) function, allocatable cm_get_cmd (this)

  *Gets the GNUPLOT command string to represent this colormap object.*
- character(len=:) function, allocatable rcm_get_clr (this)

  *Gets the GNUPLOT string defining the color distribution.*
- character(len=:) function, allocatable hcm_get_clr (this)

  *Gets the GNUPLOT string defining the color distribution.*
- character(len=:) function, allocatable ccm_get_clr (this)

  *Gets the GNUPLOT string defining the color distribution.*

### Variables

- integer(int32), parameter, public gnuplot_terminal_win32 = 1

  *Defines a Win32 terminal.*
- integer(int32), parameter, public gnuplot_terminal_wxt = 2

  *Defines a WXT terminal.*
- integer(int32), parameter, public gnuplot_terminal_qt = 3

  *Defines a QT terminal.*

- integer(int32), parameter, public gnuplot_terminal_png = 4

  *Defines a PNG terminal.*
- integer(int32), parameter, public marker_plus = 1

  *Defines a + data point marker.*
- integer(int32), parameter, public marker_x = 2

  *Defines an x data point marker.*
- integer(int32), parameter, public marker_asterisk = 3

  *Defines an ∗ data point marker.*
- integer(int32), parameter, public marker_empty_square = 4

  *Defines an empty square-shaped data point marker.*
- integer(int32), parameter, public marker_filled_square = 5

  *Defines an filled square-shaped data point marker.*
- integer(int32), parameter, public marker_empty_circle = 6

  *Defines an empty circle-shaped data point marker.*
- integer(int32), parameter, public marker_filled_circle = 7

  *Defines an filled circle-shaped data point marker.*
- integer(int32), parameter, public marker_empty_triangle = 8

  *Defines an empty triangle-shaped data point marker.*
- integer(int32), parameter, public marker_filled_triangle = 9

  *Defines an filled triangle-shaped data point marker.*
- integer(int32), parameter, public marker_empty_nabla = 10

  *Defines an empty nabla-shaped data point marker.*
- integer(int32), parameter, public marker_filled_nabla = 11

  *Defines an filled nabla-shaped data point marker.*
- integer(int32), parameter, public marker_empty_rhombus = 12

  *Defines an empty rhombus-shaped data point marker.*
- integer(int32), parameter, public marker_filled_rhombus = 13

  *Defines an filled rhombus-shaped data point marker.*
- integer(int32), parameter, public line_solid = 1

  *Defines a solid line.*
- integer(int32), parameter, public line_dashed = 2

  *Defines a dashed line.*
- integer(int32), parameter, public line_dotted = 3

  *Defines a dotted line.*
- integer(int32), parameter, public line_dash_dotted = 4

  *Defines a dash-dotted line.*
- integer(int32), parameter, public line_dash_dot_dot = 5

  *Defines a dash-dot-dotted line.*
- character(len=∗), parameter, public legend_top = "top"

  *Defines the legend should be placed at the top of the plot.*
- character(len=∗), parameter, public legend_center = "center"

  *Defines the legend should be centered on the plot.*
- character(len=∗), parameter, public legend_left = "left"

  *Defines the legend should be placed at the left of the plot.*
- character(len=∗), parameter, public legend_right = "right"

  *Defines the legend should be placed at the right of the plot.*
- character(len=∗), parameter, public legend_bottom = "bottom"

  *Defines the legend should be placed at the bottom of the plot.*
- integer(int32), parameter, public plotdata_max_name_length = 128

  *Defines the maximum number of characters allowed in a graph label.*
- integer(int32), parameter gnuplot_default_window_width = 640

*The default GNUPLOT window width, in pixels.*

- integer(int32), parameter gnuplot_default_window_height = 420

  *The default GNUPLOT window height, in pixels.*

- integer(int32), parameter gnuplot_max_label_length = 128

  *Defines the maximum number of characters allowed in a graph label.*

- character(len=∗), parameter gnuplot_default_fontname = "Calibri"

  *Defines the default font used by text on the graph.*

- integer(int32), parameter gnuplot_default_font_size = 10

  *Defines the default font size used by text on the graph.*

- integer(int32), parameter gnuplot_max_path_length = 256

  *Defines the maximum number of characters allowed in a file path.*

- type(color), parameter, public clr_black = color(0, 0, 0)

  *Defines a black color.*

- type(color), parameter, public clr_white = color(255, 255, 255)

  *Defines a white color.*

- type(color), parameter, public clr_red = color(255, 0, 0)

  *Defines a red color.*

- type(color), parameter, public clr_lime = color(0, 255, 0)

  *Defines a lime color.*

- type(color), parameter, public clr_blue = color(0, 0, 255)

  *Defines a blue color.*

- type(color), parameter, public clr_yellow = color(255, 255, 0)

  *Defines a yellow color.*

- type(color), parameter, public clr_cyan = color(0, 255, 255)

  *Defines a cyan color.*

- type(color), parameter, public clr_magenta = color(255, 0, 255)

  *Defines a magenta color.*

- type(color), parameter, public clr_silver = color(192, 192, 192)

  *Defines a silver color.*

- type(color), parameter, public clr_gray = color(128, 128, 128)

  *Defines a gray color.*

- type(color), parameter, public clr_maroon = color(128, 0, 0)

  *Defines a maroon color.*

- type(color), parameter, public clr_olive = color(128, 128, 0)

  *Defines a olive color.*

- type(color), parameter, public clr_green = color(0, 128, 0)

  *Defines a green color.*

- type(color), parameter, public clr_purple = color(128, 0, 128)

  *Defines a purple color.*

- type(color), parameter, public clr_teal = color(0, 128, 128)

  *Defines a teal color.*

- type(color), parameter, public clr_navy = color(0, 0, 128)

  *Defines a navy color.*

### 4.1.1 Detailed Description

## fplot_core

**Purpose**

Provides types and routines specific necessary to support plotting operations.

#### 4.1.2 Function/Subroutine Documentation

##### 4.1.2.1 character(len = :) function, allocatable fplot_core::ccm_get_clr ( class(cool_colormap), intent(in) *this* ) [private]

Gets the GNUPLOT string defining the color distribution.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The cool_colormap object. |

**Returns**

The command string.

Definition at line 4443 of file fplot_core.f90.

##### 4.1.2.2 subroutine fplot_core::clr_copy_from ( class(color), intent(inout) *this,* class(color), intent(in) *clr* ) [private]

Copies another color to this color.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The color object. |
| in | *clr* | The color to copy. |

Definition at line 1107 of file fplot_core.f90.

##### 4.1.2.3 pure character(6) function fplot_core::clr_to_hex_string ( class(color), intent(in) *this* ) [private]

Returns the color in hexadecimal format.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The color object. |

**Returns**

A string containing the hexadecimal equivalent.

Definition at line 1062 of file fplot_core.f90.

##### 4.1.2.4 character(len = :) function, allocatable fplot_core::cm_get_cmd ( class(colormap), intent(in) *this* ) [private]

Gets the GNUPLOT command string to represent this colormap object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The colormap object. |

**Returns**

The command string.

Definition at line 4389 of file fplot_core.f90.

#### 4.1.2.5 character(len = :) function, allocatable fplot_core::hcm_get_clr ( class(hot_colormap), intent(in) *this* ) [private]

Gets the GNUPLOT string defining the color distribution.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The hot_colormap object. |

**Returns**

The command string.

Definition at line 4430 of file fplot_core.f90.

#### 4.1.2.6 pure logical function fplot_core::leg_get_box ( class(legend), intent(in) *this* ) [private]

Gets a value determining if the legend should have a border.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The legend object. |

**Returns**

The logical value.

Definition at line 1752 of file fplot_core.f90.

#### 4.1.2.7 character(len = :) function, allocatable fplot_core::leg_get_command_txt ( class(legend), intent(in) *this* ) [private]

Gets the command string defining the legend properties.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The legend object. |

**Returns**

The GNUPLOT command string.

Definition at line 1850 of file fplot_core.f90.

#### 4.1.2.8 pure character(len = :) function, allocatable fplot_core::leg_get_horz_pos ( class(legend), intent(in) *this* ) [private]

Gets the horizontal position of the legend.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The legend object. |

**Returns**

The horizontal position of the legend (LEGEND_LEFT, LEGEND_CENTER, or LEGEND_RIGHT).

Definition at line 1775 of file fplot_core.f90.

#### 4.1.2.9 pure logical function fplot_core::leg_get_inside ( class(legend), intent(in) *this* ) [private]

Gets a value determining if the legend should be drawn inside the axes border (true), or outside the axes border (false).

**Parameters**

| | | |
|---|---|---|
| in | *this* | The legend object. |

**Returns**

The logical value.

Definition at line 1729 of file fplot_core.f90.

#### 4.1.2.10 pure character(len = :) function, allocatable fplot_core::leg_get_vert_pos ( class(legend), intent(in) *this* ) [private]

Gets the vertical position of the legend.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The legend object. |

**Returns**

The vertical position of the legend (LEGEND_TOP, LEGEND_CENTER, or LEGEND_BOTTOM).

Definition at line 1802 of file fplot_core.f90.

#### 4.1.2.11 pure logical function fplot_core::leg_get_visible ( class(legend), intent(in) *this* ) [private]

Gets a value determining if the legend is visible.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The legend object. |

**Returns**

The logical value.

Definition at line 1828 of file fplot_core.f90.

#### 4.1.2.12 subroutine fplot_core::leg_set_box ( class(legend), intent(inout) *this,* logical, intent(in) *x* ) [private]

Sets a value determining if the legend should have a border.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The legend object. |
| in | *x* | The logical value. |

Definition at line 1763 of file fplot_core.f90.

#### 4.1.2.13 subroutine fplot_core::leg_set_horz_pos ( class(legend), intent(inout) *this,* character(len = ∗), intent(in) *x* ) [private]

Sets the horizontal position of the legend.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The legend object. |
| | *x* | The horizontal position of the legend. The parameter must be set to one of the following: LEGEND_LEFT, LEGEND_CENTER, or LEGEND_RIGHT. If not, the default LEGEND_RIGHT will be used. |

Definition at line 1788 of file fplot_core.f90.

#### 4.1.2.14 subroutine fplot_core::leg_set_inside ( class(legend), intent(inout) *this,* logical, intent(in) *x* ) [private]

Sets a value determining if the legend should be drawn inside the axes border (true), or outside the axes border (false).

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The legend object. |
| in | *x* | The logical value. |

Definition at line 1741 of file fplot_core.f90.

#### 4.1.2.15 subroutine fplot_core::leg_set_vert_pos ( class(legend), intent(inout) *this*, character(len = *), intent(in) *x* ) [private]

Sets the vertical position of the legend.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The legend object. |
| | *x* | The vertical position of the legend. The parameter must be set to one of the following: LEGEND_TOP, LEGEND_CENTER, or LEGEND_BOTTOM. If not, the default LEGEND_TOP will be used. |

Definition at line 1815 of file fplot_core.f90.

#### 4.1.2.16 subroutine fplot_core::leg_set_visible ( class(legend), intent(inout) *this*, logical, intent(in) *x* ) [private]

Sets a value determining if the legend is visible.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The legend object. |
| in | *x* | The logical value. |

Definition at line 1839 of file fplot_core.f90.

#### 4.1.2.17 subroutine fplot_core::p2d_clean_up ( type(plot_2d), intent(inout) *this* ) [private]

Cleans up resources held by the plot_2d object.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_2d object. |

Definition at line 2719 of file fplot_core.f90.

#### 4.1.2.18 character(len = :) function, allocatable fplot_core::p2d_get_cmd ( class(plot_2d), intent(in) *this* ) [private]

Gets the GNUPLOT command string to represent this plot_2d object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_2d object. |

**Returns**

The command string.

Definition at line 2800 of file fplot_core.f90.

#### 4.1.2.19 pure logical function fplot_core::p2d_get_use_y2 ( class(plot_2d), intent(in) *this* ) [private]

Gets a flag determining if the secondary y-axis should be displayed.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_2d object. |

**Returns**

Returns true if the axis should be displayed; else, false.

Definition at line 2970 of file fplot_core.f90.

#### 4.1.2.20 class(plot_axis) function, pointer fplot_core::p2d_get_x_axis ( class(plot_2d), intent(in) *this* ) [private]

Gets the x-axis object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_2d object. |

**Returns**

A pointer to the x-axis object.

Definition at line 2936 of file fplot_core.f90.

#### 4.1.2.21 class(plot_axis) function, pointer fplot_core::p2d_get_y2_axis ( class(plot_2d), intent(in) *this* ) [private]

Gets the secondary y-axis object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_2d object. |

**Returns**

A pointer to the secondary y-axis object.

Definition at line 2958 of file fplot_core.f90.

#### 4.1.2.22 class(plot_axis) function, pointer fplot_core::p2d_get_y_axis ( class(plot_2d), intent(in) *this* ) [private]

Gets the y-axis object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_2d object. |

**Returns**

A pointer to the y-axis object.

Definition at line 2947 of file fplot_core.f90.

#### 4.1.2.23 subroutine fplot_core::p2d_init ( class(plot_2d), intent(inout) *this*, integer(int32), intent(in), optional *term*, class(errors), intent(inout), optional, target *err* ) [private]

Initializes the plot_2d object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_2d object. |
| in | *term* | An optional input that is used to define the terminal. The default terminal is a WXT terminal. The acceptable inputs are:<br>• GNUPLOT_TERMINAL_PNG<br>• GNUPLOT_TERMINAL_QT<br>• GNUPLOT_TERMINAL_WIN32<br>• GNUPLOT_TERMINAL_WXT |
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available. |

Definition at line 2752 of file fplot_core.f90.

#### 4.1.2.24 subroutine fplot_core::p2d_set_use_y2 ( class(plot_2d), intent(inout) *this*, logical, intent(in) *x* ) [private]

Sets a flag determining if the secondary y-axis should be displayed.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_2d object. |
| in | *x* | Set to true if the axis should be displayed; else, false. |

Definition at line 2982 of file fplot_core.f90.

#### 4.1.2.25 subroutine fplot_core::p3d_clean_up ( type(plot_3d), intent(inout) *this* ) [private]

Cleans up resources held by the plot_3d object.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_3d object. |

Definition at line 3527 of file fplot_core.f90.

#### 4.1.2.26 pure real(real64) function fplot_core::p3d_get_azimuth ( class(plot_3d), intent(in) *this* ) [private]

Gets the plot azimuth angle.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_3d object. |

**Returns**

The azimuth angle, in degrees.

Definition at line 3795 of file fplot_core.f90.

#### 4.1.2.27 character(len = :) function, allocatable fplot_core::p3d_get_cmd ( class(plot_3d), intent(in) *this* ) [private]

Gets the GNUPLOT command string to represent this plot_3d object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_3d object. |

**Returns**

The command string.

Definition at line 3608 of file fplot_core.f90.

#### 4.1.2.28 pure real(real64) function fplot_core::p3d_get_elevation ( class(plot_3d), intent(in) *this* ) [private]

Gets the plot elevation angle.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_3d object. |

**Returns**

The elevation angle, in degrees.

Definition at line 3773 of file fplot_core.f90.

#### 4.1.2.29 class(plot_axis) function, pointer fplot_core::p3d_get_x_axis ( class(plot_3d), intent(in) *this* ) [private]

Gets the x-axis object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_3d object. |

**Returns**

A pointer to the x-axis object.

Definition at line 3740 of file fplot_core.f90.

#### 4.1.2.30 class(plot_axis) function, pointer fplot_core::p3d_get_y_axis ( class(plot_3d), intent(in) *this* ) [private]

Gets the y-axis object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_3d object. |

**Returns**

A pointer to the y-axis object.

Definition at line 3751 of file fplot_core.f90.

#### 4.1.2.31 class(plot_axis) function, pointer fplot_core::p3d_get_z_axis ( class(plot_3d), intent(in) *this* ) [private]

Gets the z-axis object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_3d object. |

**Returns**

A pointer to the z-axis object.

Definition at line 3762 of file fplot_core.f90.

#### 4.1.2.32 pure logical function fplot_core::p3d_get_z_axis_intersect ( class(plot_3d), intent(in) *this* ) [private]

Gets a value determining if the z-axis should intersect the x-y plane.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_3d object. |

**Returns**

Returns true if the z-axis should intersect the x-y plane; else, false to allow the z-axis to float.

Definition at line 3819 of file fplot_core.f90.

#### 4.1.2.33 subroutine fplot_core::p3d_init ( class(plot_3d), intent(inout) *this*, integer(int32), intent(in), optional *term*, class(errors), intent(inout), optional, target *err* ) [private]

Initializes the plot_3d object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_3d object. |
| in | *term* | An optional input that is used to define the terminal. The default terminal is a WXT terminal. The acceptable inputs are:<br>• GNUPLOT_TERMINAL_PNG<br>• GNUPLOT_TERMINAL_QT<br>• GNUPLOT_TERMINAL_WIN32<br>• GNUPLOT_TERMINAL_WXT |
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available. |

Definition at line 3560 of file fplot_core.f90.

#### 4.1.2.34 subroutine fplot_core::p3d_set_azimuth ( class(plot_3d), intent(inout) *this*, real(real64), intent(in) *x* ) [private]

Sets the plot azimuth angle.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_3d object. |
| in | *x* | The azimuth angle, in degrees. |

Definition at line 3806 of file fplot_core.f90.

#### 4.1.2.35 subroutine fplot_core::p3d_set_elevation ( class(plot_3d), intent(inout) *this*, real(real64), intent(in) *x* ) [private]

Sets the plot elevation angle.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_3d object. |
| in | *x* | The elevation angle, in degrees. |

Definition at line 3784 of file fplot_core.f90.

#### 4.1.2.36 subroutine fplot_core::p3d_set_z_axis_intersect ( class(plot_3d), intent(inout) *this*, logical, intent(in) *x* ) [private]

Sets a value determining if the z-axis should intersect the x-y plane.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_3d object. |
| in | *x* | Set to true if the z-axis should intersect the x-y plane; else, false to allow the z-axis to float. |

Definition at line 3832 of file fplot_core.f90.

#### 4.1.2.37 pure logical function fplot_core::pa_get_autoscale ( class(plot_axis), intent(in) *this* ) [private]

Gets a logical value determining if the axis should be automatically scaled to fit the data.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_axis object. |

**Returns**

Returns true if the axis should be automatically scaled; else, false.

Definition at line 1528 of file fplot_core.f90.

#### 4.1.2.38 pure real(real64) function, dimension(2) fplot_core::pa_get_axis_limits ( class(plot_axis), intent(in) *this* ) [private]

Gets the axis display limits, assuming autoscaling is not active for this axis.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_axis object. |

**Returns**

A two-element array containing the limits as follows: [lower, upper].

Definition at line 1554 of file fplot_core.f90.

#### 4.1.2.39 character(len = :) function, allocatable fplot_core::pa_get_cmd_string ( class(plot_axis), intent(in) *this* ) [private]

Returns the appropriate GNUPLOT command string to define the plot_axis properties.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_axis object. |

**Returns**

The GNUPLOT command string.

Definition at line 1604 of file fplot_core.f90.

#### 4.1.2.40 pure logical function fplot_core::pa_get_log_scale ( class(plot_axis), intent(in) *this* ) [private]

Gets a logical value defining if the axis should be log scaled.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_axis object. |

**Returns**

Returns true if log scaling is applied to the axis; else, false.

Definition at line 1580 of file fplot_core.f90.

#### 4.1.2.41 pure character(len = :) function, allocatable fplot_core::pa_get_title ( class(plot_axis), intent(in) *this* ) [private]

Gets the axis' title.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_axis object. |

**Returns**

The title.

Definition at line 1476 of file fplot_core.f90.

#### 4.1.2.42 pure logical function fplot_core::pa_get_zero_axis ( class(plot_axis), intent(in) *this* ) [private]

Gets a value determining if the axis should be drawn through zero of opposing axes.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_axis object. |

**Returns**

Returns true to draw as a zero axis; else, set to false.

Definition at line 1679 of file fplot_core.f90.

#### 4.1.2.43 pure real(real32) function fplot_core::pa_get_zero_axis_width ( class(plot_axis), intent(in) *this* ) [private]

Gets the width of the line used to represent the zero axis line, if active.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_axis object. |

**Returns**

The width of the line, in pixels.

Definition at line 1703 of file fplot_core.f90.

#### 4.1.2.44 pure logical function fplot_core::pa_has_title ( class(plot_axis), intent(in) *this* ) [private]

Gets a value determining if a title has been defined for the plot_axis object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_axis object. |

**Returns**

Returns true if a title has been defined for this axis; else, returns false.

Definition at line 1515 of file fplot_core.f90.

#### 4.1.2.45 subroutine fplot_core::pa_set_autoscale ( class(plot_axis), intent(inout) *this*, logical, intent(in) *x* ) [private]

Sets a logical value determining if the axis should be automatically scaled to fit the data.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_axis object. |
| in | *x* | Set to true if the axis should be automatically scaled; else, false. |

Definition at line 1541 of file fplot_core.f90.

#### 4.1.2.46 subroutine fplot_core::pa_set_axis_limits ( class(plot_axis), intent(inout) *this*, real(real64), intent(in) *lower*, real(real64), intent(in) *upper* ) [private]

Sets the axis display limits, assuming autoscaling is not active for this axis.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_axis object. |
| in | *lower* | The lower display limit. |
| in | *upper* | The upper display limit. |

Definition at line 1568 of file fplot_core.f90.

#### 4.1.2.47 subroutine fplot_core::pa_set_log_scale ( class(plot_axis), intent(inout) *this,* logical, intent(in) *x* ) [private]

Sets a logical value defining if the axis should be log scaled.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_axis object. |
| in | *x* | Set to true if log scaling is applied to the axis; else, false. |

Definition at line 1592 of file fplot_core.f90.

#### 4.1.2.48 subroutine fplot_core::pa_set_title ( class(plot_axis), intent(inout) *this,* character(len = *), intent(in) *txt* ) [private]

Sets the axis' title.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_axis object. |
| in | *txt* | The axis title. The number of characters must be less than or equal to PLOTDATA_MAX_NAME_LENGTH; else, the text string is truncated. |

Definition at line 1489 of file fplot_core.f90.

#### 4.1.2.49 subroutine fplot_core::pa_set_zero_axis ( class(plot_axis), intent(inout) *this,* logical, intent(in) *x* ) [private]

Sets a value determining if the axis should be drawn through zero of opposing axes.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_axis object. |
| in | *x* | Set to true to draw as a zero axis; else, set to false. |

Definition at line 1691 of file fplot_core.f90.

#### 4.1.2.50 subroutine fplot_core::pa_set_zero_axis_width ( class(plot_axis), intent(inout) *this,* real(real32), intent(in) *x* ) [private]

Gets the width of the line used to represent the zero axis line, if active.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_axis object. |
| in | *x* | The width of the line, in pixels. |

Definition at line 1715 of file fplot_core.f90.

#### 4.1.2.51 character(len = :) function, allocatable fplot_core::pd2d_get_axes_cmd ( class(plot_data_2d), intent(in) *this* ) [private]

Gets the GNUPLOT command string defining which axes the data is to be plotted against.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_data_2d object. |

**Returns**

The command string.

Definition at line 3048 of file fplot_core.f90.

#### 4.1.2.52 character(len = :) function, allocatable fplot_core::pd2d_get_data_cmd ( class(plot_data_2d), intent(in) *this* ) [private]

Gets the GNUPLOT command string containing the actual data to plot.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_data_2d object. |

**Returns**

The command string.

Definition at line 3067 of file fplot_core.f90.

#### 4.1.2.53 pure integer(int32) function fplot_core::pd2d_get_data_count ( class(plot_data_2d), intent(in) *this* ) [private]

Gets the number of data points.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_data_2d object. |

**Returns**

The number of data points.

Definition at line 3100 of file fplot_core.f90.

#### 4.1.2.54 pure logical function fplot_core::pd2d_get_draw_against_y2 ( class(plot_data_2d), intent(in) *this* ) [private]

Gets a value determining if the data should be plotted against the secondary y-axis.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_data_2d object. |

**Returns**

Returns true if the data should be plotted against the secondary y-axis; else, false to plot against the primary y-axis.

Definition at line 3236 of file fplot_core.f90.

#### 4.1.2.55 pure real(real64) function fplot_core::pd2d_get_x_data ( class(plot_data_2d), intent(in) *this*, integer(int32), intent(in) *index* ) [private]

Gets the requested X data point.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_data_2d object. |
| in | *index* | The index of the data point to retrieve. |

**Returns**

The requested data point.

Definition at line 3116 of file fplot_core.f90.

#### 4.1.2.56 pure real(real64) function fplot_core::pd2d_get_y_data ( class(plot_data_2d), intent(in) *this*, integer(int32), intent(in) *index* ) [private]

Gets the requested Y data point.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_data_2d object. |
| in | *index* | The index of the data point to retrieve. |

**Returns**

The requested data point.

Definition at line 3148 of file fplot_core.f90.

#### 4.1.2.57 subroutine fplot_core::pd2d_set_data_1 ( class(plot_data_2d), intent(inout) *this*, real(real64), dimension(:), intent(in) *x*, real(real64), dimension(:), intent(in) *y*, class(errors), intent(inout), optional, target *err* ) [private]

Defines the data set.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_data_2d object. |
| in | *x* | An N-element array containing the x coordinate data. |
| in | *y* | An N-element array containing the y coordinate data. |
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available.<br>• PLOT_ARRAY_SIZE_MISMATCH_ERROR: Occurs if x and y are not the same size. |

Definition at line 3188 of file fplot_core.f90.

#### 4.1.2.58 subroutine fplot_core::pd2d_set_data_2 ( class(plot_data_2d), intent(inout) *this*, real(real64), dimension(:), intent(in) *y*, class(errors), intent(inout), optional, target *err* ) [private]

Defines the data set.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_data_2d object. |
| in | *y* | An N-element array containing the y-coordinate data. This data will be plotted against its own index. |
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available. |

Definition at line 3267 of file fplot_core.f90.

#### 4.1.2.59 subroutine fplot_core::pd2d_set_draw_against_y2 ( class(plot_data_2d), intent(inout) *this*, logical, intent(in) *x* ) [private]

Sets a value determining if the data should be plotted against the secondary y-axis.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_data_2d object. |
| in | *x* | Set to true if the data should be plotted against the secondary y-axis; else, false to plot against the primary y-axis. |

Definition at line 3249 of file fplot_core.f90.

#### 4.1.2.60 subroutine fplot_core::pd2d_set_x_data ( class(plot_data_2d), intent(inout) *this*, integer(int32), intent(in) *index*, real(real64), intent(in) *x* ) [private]

Sets the requested X data point.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_data_2d object. |
| in | *index* | The index of the data point to replace. |
| in | *x* | The data point. |

Definition at line 3133 of file fplot_core.f90.

#### 4.1.2.61 subroutine fplot_core::pd2d_set_y_data ( class(plot_data_2d), intent(inout) *this*, integer(int32), intent(in) *index*, real(real64), intent(in) *x* ) [private]

Sets the requested Y data point.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_data_2d object. |
| in | *index* | The index of the data point to replace. |
| in | *x* | The data point. |

Definition at line 3165 of file fplot_core.f90.

#### 4.1.2.62 character(len = :) function, allocatable fplot_core::pd3d_get_axes_cmd ( class(plot_data_3d), intent(in) *this* ) [private]

Gets the GNUPLOT command string defining which axes the data is to be plotted against.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_data_3d object. |

**Returns**

The command string.

Definition at line 3419 of file fplot_core.f90.

#### 4.1.2.63 character(len = :) function, allocatable fplot_core::pd3d_get_data_cmd ( class(plot_data_3d), intent(in) *this* ) [private]

Gets the GNUPLOT command string containing the actual data to plot.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_data_3d object. |

**Returns**

The command string.

Definition at line 3434 of file fplot_core.f90.

#### 4.1.2.64 pure integer(int32) function fplot_core::pd3d_get_data_count ( class(plot_data_3d), intent(in) *this* ) [private]

Gets the number of data points.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_data_3d object. |

**Returns**

The number of data points.

Definition at line 3307 of file fplot_core.f90.

#### 4.1.2.65 pure real(real64) function fplot_core::pd3d_get_x_data ( class(plot_data_3d), intent(in) *this*, integer(int32), intent(in) *index* ) [private]

Gets the requested X data point.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_data_3d object. |
| in | *index* | The index of the data point to retrieve. |

**Returns**

The requested data point.

Definition at line 3323 of file fplot_core.f90.

#### 4.1.2.66 pure real(real64) function fplot_core::pd3d_get_y_data ( class(plot_data_3d), intent(in) *this*, integer(int32), intent(in) *index* ) [private]

Gets the requested Y data point.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_data_3d object. |
| in | *index* | The index of the data point to retrieve. |

**Returns**

The requested data point.

Definition at line 3355 of file fplot_core.f90.

#### 4.1.2.67 pure real(real64) function fplot_core::pd3d_get_z_data ( class(plot_data_3d), intent(in) *this*, integer(int32), intent(in) *index* ) [private]

Gets the requested Z data point.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_data_3d object. |
| in | *index* | The index of the data point to retrieve. |

**Returns**

The requested data point.

Definition at line 3387 of file fplot_core.f90.

#### 4.1.2.68 subroutine fplot_core::pd3d_set_data_1 ( class(plot_data_3d), intent(inout) *this*, real(real64), dimension(:), intent(in) *x*, real(real64), dimension(:), intent(in) *y*, real(real64), dimension(:), intent(in) *z*, class(errors), intent(inout), optional, target *err* ) [private]

Defines the data set.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_data_2d object. |
| in | *x* | An N-element array containing the x coordinate data. |
| in | *y* | An N-element array containing the y coordinate data. |
| in | *z* | An N-element array containing the z coordinate data. |
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available.<br>• PLOT_ARRAY_SIZE_MISMATCH_ERROR: Occurs if x, y, and z are not the same size. |

Definition at line 3479 of file fplot_core.f90.

#### 4.1.2.69 subroutine fplot_core::pd3d_set_x_data ( class(plot_data_3d), intent(inout) *this*, integer(int32), intent(in) *index*, real(real64), intent(in) *x* ) [private]

Sets the requested X data point.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_data_3d object. |
| in | *index* | The index of the data point to replace. |
| in | *x* | The data point. |

Definition at line 3340 of file fplot_core.f90.

#### 4.1.2.70 subroutine fplot_core::pd3d_set_y_data ( class(plot_data_3d), intent(inout) *this*, integer(int32), intent(in) *index*, real(real64), intent(in) *x* ) [private]

Sets the requested Y data point.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_data_3d object. |
| in | *index* | The index of the data point to replace. |
| in | *x* | The data point. |

Definition at line 3372 of file fplot_core.f90.

#### 4.1.2.71 subroutine fplot_core::pd3d_set_z_data ( class(plot_data_3d), intent(inout) *this*, integer(int32), intent(in) *index*, real(real64), intent(in) *x* ) [private]

Sets the requested Z data point.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_data_3d object. |
| in | *index* | The index of the data point to replace. |
| in | *x* | The data point. |

Definition at line 3404 of file fplot_core.f90.

#### 4.1.2.72 pure character(len = :) function, allocatable fplot_core::pd_get_name ( class(plot_data), intent(in) *this* ) [private]

Gets the name to associate with this data set.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot_data object. |

**Returns**

The name.

Definition at line 1447 of file fplot_core.f90.

#### 4.1.2.73 subroutine fplot_core::pd_set_name ( class(plot_data), intent(inout) *this*, character(len = *), intent(in) *txt* ) [private]

Sets the name to associate with this data set.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_data object. |
| in | *txt* | The name. |

Definition at line 1458 of file fplot_core.f90.

#### 4.1.2.74 subroutine fplot_core::plt_clean_up ( class(plot), intent(inout) *this* ) [private]

Cleans up resources held by the plot object.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot object. |

Definition at line 1896 of file fplot_core.f90.

#### 4.1.2.75 subroutine fplot_core::plt_clear_all ( class(plot), intent(inout) *this* ) [private]

Removes all plot_data objects from the plot.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot object. |

Definition at line 2082 of file fplot_core.f90.

#### 4.1.2.76 subroutine fplot_core::plt_draw ( class(plot), intent(in) *this*, logical, intent(in), optional *persist*, class(errors), intent(inout), optional, target *err* ) [private]

Launches GNUPLOT and draws the plot per the current state of the command list.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot object. |
| in | *persist* | An optional parameter that can be used to keep GNUPLOT open. Set to true to force GNUPLOT to remain open; else, set to false to allow GNUPLOT to close after drawing. The default is true. |

**Parameters**

| | | |
|---|---|---|
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_GNUPLOT_FILE_ERROR: Occurs if the command file cannot be written. |

Definition at line 2175 of file fplot_core.f90.

#### 4.1.2.77 class(plot_data) function, pointer fplot_core::plt_get ( class(plot), intent(in) *this*, integer(int32), intent(in) *i* ) [private]

Gets a pointer to the requested plot_data object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot object. |
| in | *i* | The index of the plot_data object. |

**Returns**

A pointer to the requested plot_data object.

Definition at line 2093 of file fplot_core.f90.

#### 4.1.2.78 pure integer(int32) function fplot_core::plt_get_count ( class(plot), intent(in) *this* ) [private]

Gets the number of stored plot_data objects.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot object. |

**Returns**

The number of plot_data objects.

Definition at line 2042 of file fplot_core.f90.

#### 4.1.2.79 pure logical function fplot_core::plt_get_draw_border ( class(plot), intent(in) *this* ) [private]

Gets a value determining if the border should be drawn.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot object. |

**Returns**

Returns true if the border should be drawn; else, false.

Definition at line 2357 of file fplot_core.f90.

#### 4.1.2.80 character(len = :) function, allocatable fplot_core::plt_get_font ( class(plot), intent(in) *this* ) [private]

Gets the name of the font used for plot text.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot object. |

**Returns**

The font name.

Definition at line 2277 of file fplot_core.f90.

#### 4.1.2.81 integer(int32) function fplot_core::plt_get_font_size ( class(plot), intent(in) *this* ) [private]

Gets the size of the font used by the plot.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot object. |

**Returns**

The size of the font, in points.

Definition at line 2303 of file fplot_core.f90.

#### 4.1.2.82 type(legend) function, pointer fplot_core::plt_get_legend ( class(plot), intent(in) *this* ) [private]

Gets the plot's legend object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot object. |

**Returns**

A pointer to the legend object.

Definition at line 2031 of file fplot_core.f90.

#### 4.1.2.83 pure logical function fplot_core::plt_get_show_grid ( class(plot), intent(in) *this* ) [private]

Gets a flag determining if the grid lines should be shown.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot object. |

**Returns**

Returns true if the grid lines should be shown; else, false.

Definition at line 2144 of file fplot_core.f90.

#### 4.1.2.84 class(terminal) function, pointer fplot_core::plt_get_term ( class(plot), intent(in) *this* ) [private]

Gets the GNUPLOT terminal object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot object. |

**Returns**

A pointer to the GNUPLOT terminal object.

Definition at line 2133 of file fplot_core.f90.

#### 4.1.2.85 pure logical function fplot_core::plt_get_tics_in ( class(plot), intent(in) *this* ) [private]

Gets a value determining if the axis tic marks should point inwards.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot object. |

**Returns**

Returns true if the tic marks should point inwards; else, false if the tic marks should point outwards.

Definition at line 2333 of file fplot_core.f90.

#### 4.1.2.86 pure character(len = :) function, allocatable fplot_core::plt_get_title ( class(plot), intent(in) *this* ) [private]

Gets the plot's title.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot object. |

**Returns**

The plot's title.

Definition at line 1986 of file fplot_core.f90.

#### 4.1.2.87 pure logical function fplot_core::plt_has_title ( class(plot), intent(in) *this* ) [private]

Gets a value determining if a title has been defined for the plot object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot object. |

**Returns**

Returns true if a title has been defined for this plot; else, returns false.

Definition at line 2020 of file fplot_core.f90.

#### 4.1.2.88 subroutine fplot_core::plt_init ( class(plot), intent(inout) *this*, integer(int32), intent(in), optional *term*, class(errors), intent(inout), optional, target *err* ) [private]

Initializes the plot object.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot object. |
| in | *term* | An optional input that is used to define the terminal. The default terminal is a WXT terminal. The acceptable inputs are:<br>• GNUPLOT_TERMINAL_PNG<br>• GNUPLOT_TERMINAL_QT<br>• GNUPLOT_TERMINAL_WIN32<br>• GNUPLOT_TERMINAL_WXT |
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available. |

Definition at line 1924 of file fplot_core.f90.

#### 4.1.2.89 subroutine fplot_core::plt_pop_data ( class(plot), intent(inout) *this* ) [private]

Pops the last plot_data object from the stack.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot object. |

Definition at line 2073 of file fplot_core.f90.

#### 4.1.2.90 subroutine fplot_core::plt_push_data ( class(plot), intent(inout) *this,* class(plot_data), intent(in) *x,* class(errors), intent(inout), optional, target *err* ) [private]

Pushes a plot_data object onto the stack.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot object. |
| in | *x* | The plot_data object. |
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available. |

Definition at line 2059 of file fplot_core.f90.

#### 4.1.2.91 subroutine fplot_core::plt_save ( class(plot), intent(in) *this,* character(len = *), intent(in) *fname,* class(errors), intent(inout), optional, target *err* ) [private]

Saves a GNUPLOT command file.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The plot object. |
| in | *fname* | The filename. |
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_GNUPLOT_FILE_ERROR: Occurs if the command file cannot be written. |

Definition at line 2239 of file fplot_core.f90.

#### 4.1.2.92 subroutine fplot_core::plt_set ( class(plot), intent(inout) *this,* integer(int32), intent(in) *i,* class(plot_data), intent(in) *x* ) [private]

Sets the requested plot_data object into the plot.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot object. |
| in | *i* | The index of the plot_data object. |
| in | *x* | The plot_data object. |

Definition at line 2121 of file fplot_core.f90.

#### 4.1.2.93 subroutine fplot_core::plt_set_draw_border ( class(plot), intent(inout) *this,* logical, intent(in) *x* ) [private]

Sets a value determining if the border should be drawn.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot object. |
| in | *x* | Set to true if the border should be drawn; else, false. |

Definition at line 2368 of file fplot_core.f90.

#### 4.1.2.94 subroutine fplot_core::plt_set_font ( class(plot), intent(inout) *this,* character(len = ∗), intent(in) *x* ) [private]

Sets the name of the font used for plot text.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot object. |
| in | *x* | The font name. |

Definition at line 2290 of file fplot_core.f90.

#### 4.1.2.95 subroutine fplot_core::plt_set_font_size ( class(plot), intent(inout) *this,* integer(int32), intent(in) *x* ) [private]

Sets the size of the font used by the plot.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot object. |
| in | *x* | The font size, in points. If a value of zero is provided, the font size is reset to its default value; or, if a negative value is provided, the absolute value of the supplied value is utilized. |

Definition at line 2318 of file fplot_core.f90.

#### 4.1.2.96 subroutine fplot_core::plt_set_show_grid ( class(plot), intent(inout) *this,* logical, intent(in) *x* ) [private]

Sets a flag determining if the grid lines should be shown.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot object. |
| in | *x* | Set to true if the grid lines should be shown; else, false. |

Definition at line 2155 of file fplot_core.f90.

#### 4.1.2.97 subroutine fplot_core::plt_set_tics_in ( class(plot), intent(inout) *this,* logical, intent(in) *x* ) [private]

Sets a value determining if the axis tic marks should point inwards.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot object. |
| in | *x* | Set to true if the tic marks should point inwards; else, false if the tic marks should point outwards. |

Definition at line 2346 of file fplot_core.f90.

#### 4.1.2.98 subroutine fplot_core::plt_set_title ( class(plot), intent(inout) *this,* character(len = *), intent(in) *txt* ) [private]

Sets the plot's title.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot object. |
| in | *txt* | The plot's title. The number of characters must be less than or equal to PLOTDATA_MAX_NAME_LENGTH; else, the text string is truncated. |

Definition at line 1999 of file fplot_core.f90.

#### 4.1.2.99 character(len = :) function, allocatable fplot_core::png_get_command_string ( class(png_terminal), intent(in) *this* ) [private]

Returns the appropriate GNUPLOT command string to establish appropriate parameters.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The terminal object. |

**Returns**

The GNUPLOT command string.

Definition at line 1411 of file fplot_core.f90.

#### 4.1.2.100 pure character(len = :) function, allocatable fplot_core::png_get_filename ( class(png_terminal), intent(in) *this* ) [private]

Gets the filename for the output PNG file.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The png_terminal object. |

**Returns**

The filename, including the file extension (.png).

Definition at line 1381 of file fplot_core.f90.

#### 4.1.2.101 pure character(len = :) function, allocatable fplot_core::png_get_term_string ( class(png_terminal), intent(in) *this* ) [private]

Retrieves a GNUPLOT terminal identifier string.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The png_terminal object. |

**Returns**

The string.

Definition at line 1370 of file fplot_core.f90.

#### 4.1.2.102 subroutine fplot_core::png_set_filename ( class(png_terminal), intent(inout) *this*, character(len = *), intent(in) *txt* ) [private]

Sets the filename for the output PNG file.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The png_terminal object. |
| in | *The* | filename, including the file extension (.png). |

Definition at line 1392 of file fplot_core.f90.

#### 4.1.2.103 pure character(len = :) function, allocatable fplot_core::qt_get_term_string ( class(qt_terminal), intent(in) *this* ) [private]

Retrieves a GNUPLOT terminal identifier string.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The qt_terminal object. |

**Returns**

The string.

Definition at line 1343 of file fplot_core.f90.

#### 4.1.2.104 character(len = :) function, allocatable fplot_core::rcm_get_clr ( class(rainbow_colormap), intent(in) *this* ) [private]

Gets the GNUPLOT string defining the color distribution.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The rainbow_colormap object. |

**Returns**

The command string.

Definition at line 4416 of file fplot_core.f90.

#### 4.1.2.105 character(len = :) function, allocatable fplot_core::spd_get_cmd ( class(scatter_plot_data), intent(in) *this* ) [private]

Gets the GNUPLOT command string to represent this scatter_plot_data object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The scatter_plot_data object. |

**Returns**

The command string.

Definition at line 2382 of file fplot_core.f90.

#### 4.1.2.106 pure logical function fplot_core::spd_get_draw_line ( class(scatter_plot_data), intent(in) *this* ) [private]

Gets a value determining if a line should be drawn.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The scatter_plot_data object. |

**Returns**

Returns true if the line should be drawn; else, false.

Definition at line 2541 of file fplot_core.f90.

#### 4.1.2.107 pure logical function fplot_core::spd_get_draw_markers ( class(scatter_plot_data), intent(in) *this* ) [private]

Gets a value determining if data point markers should be drawn.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The scatter_plot_data object. |

**Returns**

Returns true if the markers should be drawn; else, false.

Definition at line 2563 of file fplot_core.f90.

#### 4.1.2.108 pure type(color) function fplot_core::spd_get_line_color ( class(scatter_plot_data), intent(in) *this* ) [private]

Gets the line color.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The scatter_plot_data object. |

**Returns**

The color.

Definition at line 2519 of file fplot_core.f90.

#### 4.1.2.109 pure integer(int32) function fplot_core::spd_get_line_style ( class(scatter_plot_data), intent(in) *this* ) [private]

Gets the line style.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The scatter_plot_data object. |

**Returns**

The line style. The line style must be one of the following:

- LINE_DASHED
- LINE_DASH_DOTTED
- LINE_DASH_DOT_DOT
- LINE_DOTTED
- LINE_SOLID

Definition at line 2484 of file fplot_core.f90.

#### 4.1.2.110 pure real(real32) function fplot_core::spd_get_line_width ( class(scatter_plot_data), intent(in) *this* ) [private]

Gets the width of the line, in pixels.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The scatter_plot_data object. |

**Returns**

The line width.

Definition at line 2457 of file fplot_core.f90.

#### 4.1.2.111 pure integer(int32) function fplot_core::spd_get_marker_frequency ( class(scatter_plot_data), intent(in) *this* ) [private]

Gets the marker frequency.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The scatter_plot_data object. |

**Returns**

The marker frequency.

Definition at line 2672 of file fplot_core.f90.

#### 4.1.2.112 pure real(real32) function fplot_core::spd_get_marker_scaling ( class(scatter_plot_data), intent(in) *this* ) [private]

Gets the marker scaling.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The scatter_plot_data object. |

**Returns**

The scaling factor.

Definition at line 2650 of file fplot_core.f90.

#### 4.1.2.113 pure integer(int32) function fplot_core::spd_get_marker_style ( class(scatter_plot_data), intent(in) *this* ) [private]

Gets the marker style.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The scatter_plot_data object. |

**Returns**

The marker type. The marker type must be one of the following:

- MARKER_ASTERISK
- MARKER_EMPTY_CIRCLE
- MARKER_EMPTY_NABLA
- MARKER_EMPTY_RHOMBUS
- MARKER_EMPTY_SQUARE
- MARKER_EMPTY_TRIANGLE

- MARKER_FILLED_CIRCLE
- MARKER_FILLED_NABLA
- MARKER_FILLED_RHOMBUS
- MARKER_FILLED_SQUARE
- MARKER_FILLED_TRIANGLE
- MARKER_PLUS
- MARKER_X

Definition at line 2598 of file fplot_core.f90.

#### 4.1.2.114 pure logical function fplot_core::spd_get_use_auto_colors ( class(scatter_plot_data), intent(in) *this* ) [private]

Gets a value determining if GNUPLOT should automatically choose line colors.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The scatter_plot_data object. |

**Returns**

Returns true if GNUPLOT should choose colors; else, false.

Definition at line 2695 of file fplot_core.f90.

#### 4.1.2.115 subroutine fplot_core::spd_set_draw_line ( class(scatter_plot_data), intent(inout) *this,* logical, intent(in) *x* ) [private]

Sets a value determining if a line should be drawn.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The scatter_plot_data object. |
| in | *x* | Set to true if the line should be drawn; else, false. |

Definition at line 2552 of file fplot_core.f90.

#### 4.1.2.116 subroutine fplot_core::spd_set_draw_markers ( class(scatter_plot_data), intent(inout) *this,* logical, intent(in) *x* ) [private]

Sets a value determining if data point markers should be drawn.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The scatter_plot_data object. |
| in | *x* | Set to true if the markers should be drawn; else, false. |

Definition at line 2574 of file fplot_core.f90.

#### 4.1.2.117 subroutine fplot_core::spd_set_line_color ( class(scatter_plot_data), intent(inout) *this,* type(color), intent(in) *x* ) [private]

Sets the line color.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The scatter_plot_data object. |
| in | *x* | The color. |

Definition at line 2530 of file fplot_core.f90.

#### 4.1.2.118 subroutine fplot_core::spd_set_line_style ( class(scatter_plot_data), intent(inout) *this,* integer(int32), intent(in) *x* ) [private]

Sets the line style.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The scatter_plot_data object. |
| in | *x* | The line style. The line style must be one of the following:<br>• LINE_DASHED<br>• LINE_DASH_DOTTED<br>• LINE_DASH_DOT_DOT<br>• LINE_DOTTED<br>• LINE_SOLID |

Definition at line 2501 of file fplot_core.f90.

#### 4.1.2.119 subroutine fplot_core::spd_set_line_width ( class(scatter_plot_data), intent(inout) *this,* real(real32), intent(in) *x* ) [private]

Sets the width of the line, in pixels.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The scatter_plot_data object. |
| in | *x* | The line width. |

Definition at line 2468 of file fplot_core.f90.

#### 4.1.2.120 subroutine fplot_core::spd_set_marker_frequency ( class(scatter_plot_data), intent(inout) *this,* integer(int32), intent(in) *x* ) [private]

Sets the marker frequency.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The scatter_plot_data object. |
| in | *x* | The marker frequency. |

Definition at line 2683 of file fplot_core.f90.

#### 4.1.2.121 subroutine fplot_core::spd_set_marker_scaling ( class(scatter_plot_data), intent(inout) *this*, real(real32), intent(in) *x* ) [private]

Sets the marker scaling.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The scatter_plot_data object. |
| in | *x* | The scaling factor. |

Definition at line 2661 of file fplot_core.f90.

#### 4.1.2.122 subroutine fplot_core::spd_set_marker_style ( class(scatter_plot_data), intent(inout) *this*, integer(int32), intent(in) *x* ) [private]

Sets the marker style.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The scatter_plot_data object. |
| in | *x* | The marker type. The marker type must be one of the following:<br>• MARKER_ASTERISK<br>• MARKER_EMPTY_CIRCLE<br>• MARKER_EMPTY_NABLA<br>• MARKER_EMPTY_RHOMBUS<br>• MARKER_EMPTY_SQUARE<br>• MARKER_EMPTY_TRIANGLE<br>• MARKER_FILLED_CIRCLE<br>• MARKER_FILLED_NABLA<br>• MARKER_FILLED_RHOMBUS<br>• MARKER_FILLED_SQUARE<br>• MARKER_FILLED_TRIANGLE<br>• MARKER_PLUS<br>• MARKER_X |

Definition at line 2623 of file fplot_core.f90.

#### 4.1.2.123 subroutine fplot_core::spd_set_use_auto_colors ( class(scatter_plot_data), intent(inout) *this*, logical, intent(in) *x* ) [private]

Sets a value determining if GNUPLOT should automatically choose line colors.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The scatter_plot_data object. |
| in | *x* | Set to true if GNUPLOT should choose colors; else, false. |

Definition at line 2707 of file fplot_core.f90.

#### 4.1.2.124 subroutine fplot_core::surf_clean_up ( type(surface_plot), intent(inout) *this* ) [private]

Cleans up resources held by the surface_plot object.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The surface_plot object. |

Definition at line 4133 of file fplot_core.f90.

#### 4.1.2.125 character(len = :) function, allocatable fplot_core::surf_get_cmd ( class(surface_plot), intent(in) *this* ) [private]

Gets the GNUPLOT command string to represent this surface_plot object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The surface_plot object. |

**Returns**

The command string.

Definition at line 4205 of file fplot_core.f90.

#### 4.1.2.126 class(colormap) function, pointer fplot_core::surf_get_colormap ( class(surface_plot), intent(in) *this* ) [private]

Gets a pointer to the colormap object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The surface_plot object. |

**Returns**

A pointer to the colormap object. If no colormap is defined, a null pointer is returned.

Definition at line 4263 of file fplot_core.f90.

#### 4.1.2.127 pure logical function fplot_core::surf_get_show_colorbar ( class(surface_plot), intent(in) *this* ) [private]

Gets a value determining if the colorbar should be shown.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The surface_plot object. |

**Returns**

Returns true if the colorbar should be drawn; else, false.

Definition at line 4365 of file fplot_core.f90.

#### 4.1.2.128 pure logical function fplot_core::surf_get_show_contours ( class(surface_plot), intent(in) *this* ) [private]

Gets a value determining if a contour plot should be drawn in conjunction with the surface plot.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The surface_plot object. |

**Returns**

Returns true if the contour plot should be drawn; else, false to only draw the surface.

Definition at line 4341 of file fplot_core.f90.

#### 4.1.2.129 pure logical function fplot_core::surf_get_show_hidden ( class(surface_plot), intent(in) *this* ) [private]

Gets a value indicating if hidden lines should be shown.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The surface_plot object. |

**Returns**

Returns true if hidden lines should be shown; else, false.

Definition at line 4182 of file fplot_core.f90.

#### 4.1.2.130 pure logical function fplot_core::surf_get_smooth ( class(surface_plot), intent(in) *this* ) [private]

Gets a value determining if the plotted surfaces should be smoothed.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The surface_plot object. |

**Returns**

Returns true if the surface should be smoothed; else, false.

Definition at line 4316 of file fplot_core.f90.

#### 4.1.2.131 subroutine fplot_core::surf_init ( class(surface_plot), intent(inout) *this,* integer(int32), intent(in), optional *term,* class(errors), intent(inout), optional, target *err* ) [private]

Initializes the surface_plot object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The surface_plot object. |
| in | *term* | An optional input that is used to define the terminal. The default terminal is a WXT terminal. The acceptable inputs are:<br>• GNUPLOT_TERMINAL_PNG<br>• GNUPLOT_TERMINAL_QT<br>• GNUPLOT_TERMINAL_WIN32<br>• GNUPLOT_TERMINAL_WXT |
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available. |

Definition at line 4160 of file fplot_core.f90.

#### 4.1.2.132 subroutine fplot_core::surf_set_colormap ( class(surface_plot), intent(inout) *this,* class(colormap), intent(in) *x,* class(errors), intent(inout), optional, target *err* ) [private]

Sets the colormap object.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The surface_plot object. |
| in | *x* | The colormap object. Notice, a copy of this object is stored, and the surface_plot object then manages the lifetime of the copy. |
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available. |

Definition at line 4282 of file fplot_core.f90.

#### 4.1.2.133 subroutine fplot_core::surf_set_show_colorbar ( class(surface_plot), intent(inout) *this*, logical, intent(in) *x* ) [private]

Sets a value determining if the colorbar should be shown.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The surface_plot object. |
| in | *x* | Set to true if the colorbar should be drawn; else, false. |

Definition at line 4376 of file fplot_core.f90.

#### 4.1.2.134 subroutine fplot_core::surf_set_show_contours ( class(surface_plot), intent(inout) *this*, logical, intent(in) *x* ) [private]

Sets a value determining if a contour plot should be drawn in conjunction with the surface plot.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The surface_plot object. |
| in | *x* | Set to true if the contour plot should be drawn; else, false to only draw the surface. |

Definition at line 4354 of file fplot_core.f90.

#### 4.1.2.135 subroutine fplot_core::surf_set_show_hidden ( class(surface_plot), intent(inout) *this*, logical, intent(in) *x* ) [private]

Sets a value indicating if hidden lines should be shown.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The surface_plot object. |
| in | *x* | Set to true if hidden lines should be shown; else, false. |

Definition at line 4193 of file fplot_core.f90.

#### 4.1.2.136 subroutine fplot_core::surf_set_smooth ( class(surface_plot), intent(inout) *this*, logical, intent(in) *x* ) [private]

Sets a value determining if the plotted surfaces should be smoothed.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The surface_plot object. |
| in | *x* | Set to true if the surface should be smoothed; else, false. |

Definition at line 4328 of file fplot_core.f90.

#### 4.1.2.137 character(len = :) function, allocatable fplot_core::surfd_get_cmd ( class(surface_plot_data), intent(in) *this* ) [private]

Gets the GNUPLOT command string to represent this surface_plot_data object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The surface_plot_data object. |

**Returns**

The command string.

Definition at line 3990 of file fplot_core.f90.

#### 4.1.2.138 character(len = :) function, allocatable fplot_core::surfd_get_data_cmd ( class(surface_plot_data), intent(in) *this* ) [private]

Gets the GNUPLOT command string containing the actual data to plot.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The surface_plot_data object. |

**Returns**

The GNUPLOT command string.

Definition at line 4029 of file fplot_core.f90.

#### 4.1.2.139 pure integer(int32) function fplot_core::surfd_get_size ( class(surface_plot_data), intent(in) *this,* integer(int32), intent(in) *dim* ) [private]

Gets the size of the stored data set.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The suface_plot_data object. |
| in | *dim* | The dimension of interest. Notice, data is stored as a 2D matrix (i.e. only 1 and 2 are valid inputs). |

**Returns**

The size of the requested dimension.

Definition at line 3847 of file fplot_core.f90.

#### 4.1.2.140 pure logical function fplot_core::surfd_get_wireframe ( class(surface_plot_data), intent(in) *this* ) [private]

Gets a value determining if a wireframe mesh should be displayed.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The surface_plot_data object. |

**Returns**

Returns true if a wireframe mesh should be displayed; else, false to display a solid surface.

Definition at line 3966 of file fplot_core.f90.

#### 4.1.2.141 pure real(real64) function fplot_core::surfd_get_x ( class(surface_plot_data), intent(in) *this*, integer(int32), intent(in) *i*, integer(int32), intent(in) *j* ) [private]

Gets the requested X data point.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The surface_plot_data object. |
| in | *i* | The row index. |
| in | *j* | The column index. |

**Returns**

The value.

Definition at line 3865 of file fplot_core.f90.

#### 4.1.2.142 pure real(real64) function fplot_core::surfd_get_y ( class(surface_plot_data), intent(in) *this*, integer(int32), intent(in) *i*, integer(int32), intent(in) *j* ) [private]

Gets the requested Y data point.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The surface_plot_data object. |
| in | *i* | The row index. |
| in | *j* | The column index. |

**Returns**

The value.

Definition at line 3899 of file fplot_core.f90.

#### 4.1.2.143 pure real(real64) function fplot_core::surfd_get_z ( class(surface_plot_data), intent(in) *this*, integer(int32), intent(in) *i*, integer(int32), intent(in) *j* ) [private]

Gets the requested Z data point.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The surface_plot_data object. |
| in | *i* | The row index. |
| in | *j* | The column index. |

**Returns**

The value.

Definition at line 3933 of file fplot_core.f90.

#### 4.1.2.144 subroutine fplot_core::surfd_set_data_1 ( class(surface_plot_data), intent(inout) *this*, real(real64), dimension(:,:), intent(in) *x*, real(real64), dimension(:,:), intent(in) *y*, real(real64), dimension(:,:), intent(in) *z*, class(errors), intent(inout), optional, target *err* ) [private]

Defines the data set.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The plot_data_2d object. |
| in | *x* | An M-by-N matrix containing the x-coordinate data. |
| in | *y* | An M-by-N matrix containing the y-coordinate data. |
| in | *z* | An M-by-N matrix containing the z-coordinate data. |
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available.<br>• PLOT_ARRAY_SIZE_MISMATCH_ERROR: Occurs if x, y, and z are not the same size. |

Definition at line 4078 of file fplot_core.f90.

#### 4.1.2.145 subroutine fplot_core::surfd_set_wireframe ( class(surface_plot_data), intent(inout) *this*, logical, intent(in) *x* ) [private]

Sets a value determining if a wireframe mesh should be displayed.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The surface_plot_data object. |
| in | *x* | Set to true if a wireframe mesh should be displayed; else, false to display a solid surface. |

Definition at line 3978 of file fplot_core.f90.

#### 4.1.2.146 subroutine fplot_core::surfd_set_x ( class(surface_plot_data), intent(inout) *this*, integer(int32), intent(in) *i*, integer(int32), intent(in) *j*, real(real64), intent(in) *x* ) [private]

Sets the requested X data point.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The surface_plot_data object. |
| in | *i* | The row index. |
| in | *j* | The column index. |
| in | *x* | The value. |

Definition at line 3883 of file fplot_core.f90.

#### 4.1.2.147 subroutine fplot_core::surfd_set_y ( class(surface_plot_data), intent(inout) *this*, integer(int32), intent(in) *i*, integer(int32), intent(in) *j*, real(real64), intent(in) *x* ) [private]

Sets the requested Y data point.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The surface_plot_data object. |
| in | *i* | The row index. |
| in | *j* | The column index. |
| in | *x* | The value. |

Definition at line 3917 of file fplot_core.f90.

#### 4.1.2.148 subroutine fplot_core::surfd_set_z ( class(surface_plot_data), intent(inout) *this*, integer(int32), intent(in) *i*, integer(int32), intent(in) *j*, real(real64), intent(in) *x* ) [private]

Sets the requested Z data point.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The surface_plot_data object. |
| in | *i* | The row index. |
| in | *j* | The column index. |
| in | *x* | The value. |

Definition at line 3951 of file fplot_core.f90.

#### 4.1.2.149 character(len = :) function, allocatable fplot_core::term_get_command_string ( class(terminal), intent(in) *this* ) [private]

Returns the appropriate GNUPLOT command string to establish appropriate parameters.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The terminal object. |

**Returns**

The GNUPLOT command string.

Definition at line 1291 of file fplot_core.f90.

#### 4.1.2.150 pure character(len = :) function, allocatable fplot_core::term_get_font_name ( class(terminal), intent(in) *this* ) [private]

Gets the name of the font used for text displayed by the graph.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The terminal object. |

**Returns**

The font name.

Definition at line 1232 of file fplot_core.f90.

#### 4.1.2.151 pure integer function fplot_core::term_get_font_size ( class(terminal), intent(in) *this* ) [private]

Gets the size of the font used by the graph.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The terminal object. |

**Returns**

The font size, in points.

Definition at line 1262 of file fplot_core.f90.

#### 4.1.2.152 pure integer(int32) function fplot_core::term_get_plot_window_number ( class(terminal), intent(in) *this* ) [private]

Gets the targeted plot window number.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The terminal object. |

**Returns**

The plot window number.

Definition at line 1180 of file fplot_core.f90.

#### 4.1.2.153 pure character(len = :) function, allocatable fplot_core::term_get_title ( class(terminal), intent(in) *this* ) [private]

Gets the plot window's title.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The terminal object. |

**Returns**

The title.

Definition at line 1202 of file fplot_core.f90.

#### 4.1.2.154 pure integer function fplot_core::term_get_window_height ( class(terminal), intent(in) *this* ) [private]

Gets the height of the plot window.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The terminal object. |

**Returns**

The height of the plot window.

Definition at line 1151 of file fplot_core.f90.

#### 4.1.2.155 pure integer function fplot_core::term_get_window_width ( class(terminal), intent(in) *this* ) [private]

Gets the width of the plot window.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The terminal object. |

**Returns**

The width of the plot window.

Definition at line 1122 of file fplot_core.f90.

#### 4.1.2.156 subroutine fplot_core::term_set_font_name ( class(terminal), intent(inout) *this*, character(len = *), intent(in) *name* ) [private]

Sets the name of the font used for text displayed by the graph.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The terminal object. |
| in | *name* | The name of the font. If no name is supplied, the name is reset back to its default setting. |

Definition at line 1244 of file fplot_core.f90.

#### 4.1.2.157 subroutine fplot_core::term_set_font_size ( class(terminal), intent(inout) *this,* integer(int32), intent(in) *sz* ) [private]

Sets the size of the font used by the graph.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The terminal object. |
| in | *sz* | The font size, in points. If a value of zero is provided, the font size is reset to its default value; or, if a negative value is provided, the absolute value of the supplied value is utilized. |

Definition at line 1275 of file fplot_core.f90.

#### 4.1.2.158 subroutine fplot_core::term_set_plot_window_number ( class(terminal), intent(inout) *this,* integer(int32), intent(in) *x* ) [private]

Sets the targeted plot window number.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The terminal object. |
| in | *x* | The plot window number. |

Definition at line 1191 of file fplot_core.f90.

#### 4.1.2.159 subroutine fplot_core::term_set_title ( class(terminal), intent(inout) *this,* character(len = ∗), intent(in) *txt* ) [private]

Sets the plot window's title.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The terminal object. |
| in | *txt* | The title. |

Definition at line 1213 of file fplot_core.f90.

#### 4.1.2.160 subroutine fplot_core::term_set_window_height ( class(terminal), intent(inout) *this,* integer, intent(in) *x* ) [private]

Sets the height of the plot window.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The terminal object. |
| in | *x* | The height of the plot window. If a value of zero is provided, the window height is reset to its default value; or, if a negative value is provided, the absolute value of the supplied value is utilized. |

Definition at line 1165 of file fplot_core.f90.

#### 4.1.2.161 subroutine fplot_core::term_set_window_width ( class(terminal), intent(inout) *this*, integer, intent(in) *x* ) [private]

Sets the width of the plot window.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The terminal object. |
| in | *x* | The width of the plot window. If a value of zero is provided, the window width is reset to its default value; or, if a negative value is provided, the absolute value of the supplied value is utilized. |

Definition at line 1136 of file fplot_core.f90.

#### 4.1.2.162 pure character(len = :) function, allocatable fplot_core::wt_get_term_string ( class(windows_terminal), intent(in) *this* ) [private]

Retrieves a GNUPLOT terminal identifier string.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The windows_terminal object. |

**Returns**

The string.

Definition at line 1330 of file fplot_core.f90.

#### 4.1.2.163 pure character(len = :) function, allocatable fplot_core::wxt_get_term_string ( class(wxt_terminal), intent(in) *this* ) [private]

Retrieves a GNUPLOT terminal identifier string.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The wxt_terminal object. |

**Returns**

The string.

Definition at line 1357 of file fplot_core.f90.

#### 4.1.2.164 character(len = :) function, allocatable fplot_core::xa_get_id ( class(x_axis), intent(in) *this* ) [private]

Gets the axis identification string.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The x_axis object. |

**Returns**

The string.

Definition at line 2995 of file fplot_core.f90.

#### 4.1.2.165 character(len = :) function, allocatable fplot_core::y2a_get_id ( class(y2_axis), intent(in) *this* ) [private]

Gets the axis identification string.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The y2_axis object. |

**Returns**

The string.

Definition at line 3021 of file fplot_core.f90.

#### 4.1.2.166 character(len = :) function, allocatable fplot_core::ya_get_id ( class(y_axis), intent(in) *this* ) [private]

Gets the axis identification string.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The y_axis object. |

**Returns**

The string.

Definition at line 3008 of file fplot_core.f90.

#### 4.1.2.167 character(len = :) function, allocatable fplot_core::za_get_id ( class(z_axis), intent(in) *this* ) [private]

Gets the axis identification string.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The z_axis object. |

#### Returns

The string.

Definition at line 3034 of file fplot_core.f90.

## 4.2 fplot_errors Module Reference

### plot_errors

#### Variables

- integer(int32), parameter plot_out_of_memory_error = 1000

  *Occurs if there is insufficient memory available for the requested operation.*
- integer(int32), parameter plot_invalid_input_error = 1001

  *Occurs if an invalid input is provided.*
- integer(int32), parameter plot_invalid_operation_error = 1002

  *Occurs if an attempt is made to perform an invalid operation.*
- integer(int32), parameter plot_array_size_mismatch_error = 1003

  *Occurs if there is an array size mismatch error.*
- integer(int32), parameter plot_gnuplot_file_error = 1004

  *Occurs if there is a GNUPLOT file error.*

### 4.2.1 Detailed Description

### plot_errors

#### Purpose

Provides error codes for plot routines.

## 4.3 fplot_list Module Reference

### fplot_list

#### Data Types

- type container

  *A container type allowing storage of most any Fortran type.*
- type list

  *A generic list container.*

#### Functions/Subroutines

- class(*) function, pointer cntr_get_item (this)

  *Retrieves the stored data represented as a pointer to a polymorhpic type.*
- subroutine cntr_set_item (this, x)

  *Stores the specified item in the container.*
- pure integer(int32) function list_get_count (this)

  *Gets the number of items in the list.*
- pure integer(int32) function list_get_capacity (this)

  *Gets the capacity of the list.*
- subroutine list_set_capacity (this, n, err)

  *Sets the capacity of the list.*
- type(container) function list_get (this, i)

  *Gets an item from the list.*
- subroutine list_set (this, i, x)

  *Sets an item into the list.*
- subroutine list_push (this, x, err)

  *Pushes an item onto the end of the list.*
- subroutine list_pop (this)

  *Pops the last item from the end of the list.*
- subroutine list_insert (this, i, x, err)

  *Inserts an item into the list.*
- subroutine list_remove (this, i, err)

  *Removes an item from the list.*
- subroutine list_clear (this)

  *Clears the contents of the list.*

#### Variables

- integer(int32), parameter default_buffer_size = 10

  *The default buffer size.*

### 4.3.1 Detailed Description

## fplot_list

#### Purpose

This module provides a collection suitable for supporting plotting operations.

### 4.3.2 Function/Subroutine Documentation

#### 4.3.2.1 class(*) function, pointer fplot_list::cntr_get_item ( class(container), intent(in) *this* ) [private]

Retrieves the stored data represented as a pointer to a polymorhpic type.

#### Parameters

| | | |
|---|---|---|
| in | *this* | The container object. |

**Returns**

A pointer to the stored data.

Definition at line 79 of file fplot_list.f90.

#### 4.3.2.2 subroutine fplot_list::cntr_set_item ( class(container), intent(inout) *this,* class(∗), intent(in), target *x* ) [private]

Stores the specified item in the container.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The container object. |
| in | *x* | The object to store. |

Definition at line 90 of file fplot_list.f90.

#### 4.3.2.3 subroutine fplot_list::list_clear ( class(list), intent(inout) *this* ) [private]

Clears the contents of the list.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The list object. |

Definition at line 373 of file fplot_list.f90.

#### 4.3.2.4 type(container) function fplot_list::list_get ( class(list), intent(in) *this,* integer(int32), intent(in) *i* ) [private]

Gets an item from the list.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The list object. |
| in | *i* | The index of the item. |

**Returns**

A container object containing requested item.

Definition at line 191 of file fplot_list.f90.

#### 4.3.2.5 pure integer(int32) function fplot_list::list_get_capacity ( class(list), intent(in) *this* ) [private]

Gets the capacity of the list.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The list object. |

**Returns**

The capacity of the list.

Definition at line 114 of file fplot_list.f90.

#### 4.3.2.6 pure integer(int32) function fplot_list::list_get_count ( class(list), intent(in) *this* ) [private]

Gets the number of items in the list.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The list object. |

**Returns**

The number of items stored in the list.

Definition at line 103 of file fplot_list.f90.

#### 4.3.2.7 subroutine fplot_list::list_insert ( class(list), intent(inout) *this*, integer(int32), intent(in) *i*, class(*), intent(in) *x*, class(errors), intent(inout), optional, target *err* ) [private]

Inserts an item into the list.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The list object. |
| in | *i* | The index at which to insert the item. |
| in | *x* | The item to insert. |
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_INVALID_INPUT_ERROR: Occurs if i is less than or equal to 0, or if i is larger than 1 element beyond the current size of the list.<br>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available. |

Definition at line 272 of file fplot_list.f90.

#### 4.3.2.8 subroutine fplot_list::list_pop ( class(list), intent(inout) *this* ) [private]

Pops the last item from the end of the list.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The list object. |

Definition at line 247 of file fplot_list.f90.

#### 4.3.2.9 subroutine fplot_list::list_push ( class(list), intent(inout) *this,* class(∗), intent(in) *x,* class(errors), intent(inout), optional, target *err* ) [private]

Pushes an item onto the end of the list.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The list object. |
| in | *x* | The object to add to the list. |
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available. |

Definition at line 227 of file fplot_list.f90.

#### 4.3.2.10 subroutine fplot_list::list_remove ( class(list), intent(inout) *this,* integer(int32), intent(in) *i,* class(errors), intent(inout), optional, target *err* ) [private]

Removes an item from the list.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The list object. |
| in | *i* | The index of the item to remove. |
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_INVALID_INPUT_ERROR: Occurs if i is less than or equal to 0, or if i is larger than 1 element beyond the current size of the list.<br>• PLOT_INVALID_OPERATION_ERROR: Occurs if attempting to remove an item when there are no items left in the list to remove. |

Definition at line 329 of file fplot_list.f90.

#### 4.3.2.11 subroutine fplot_list::list_set ( class(list), intent(inout) *this,* integer(int32), intent(in) *i,* class(∗), intent(in) *x* ) [private]

Sets an item into the list.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The list object. |
| in | *i* | The index of the item. |
| in | *x* | The item to place into the list. |

Definition at line 204 of file fplot_list.f90.

#### 4.3.2.12 subroutine fplot_list::list_set_capacity ( class(list), intent(inout) *this,* integer(int32), intent(in) *n,* class(errors), intent(inout), optional, target *err* ) [private]

Sets the capacity of the list.

**Parameters**

| | | |
|---|---|---|
| in,out | *this* | The list object. |
| in | *n* | The desired capacity of the list. This value must not be less than the number of items already stored in the list. |
| out | *err* | An optional errors-based object that if provided can be used to retrieve information relating to any errors encountered during execution. If not provided, a default implementation of the errors class is used internally to provide error handling. Possible errors and warning messages that may be encountered are as follows.<br>• PLOT_INVALID_INPUT_ERROR: Occurs if n is less than the number of items already stored in the list.<br>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available. |

Definition at line 138 of file fplot_list.f90.

# 5 Data Type Documentation

## 5.1 fplot_core::cm_get_string_result Interface Reference

Retrieves a string from a colormap.

**Private Member Functions**

- character(len=:) function, allocatable **cm_get_string_result** (this)

### 5.1.1 Detailed Description

Retrieves a string from a colormap.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The colormap object. |

**Returns**

The string.

Definition at line 1045 of file fplot_core.f90.

The documentation for this interface was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.2 fplot_core::color Type Reference

Describes an RGB color.

#### Private Member Functions

- procedure, pass to_hex_string => clr_to_hex_string
  *Returns the color in hexadecimal format.*
- procedure, pass copy_from => clr_copy_from
  *Copies another color to this color.*

#### Private Attributes

- integer(int32) red = 0
  *The red component of the color (must be between 0 and 255).*
- integer(int32) green = 0
  *The green component of the color (must be between 0 and 255).*
- integer(int32) blue = 255
  *The blue component of the color (must be between 0 and 255).*

### 5.2.1 Detailed Description

Describes an RGB color.

Definition at line 178 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.3 fplot_core::colormap Type Reference

A colormap object for a surface plot.

Inheritance diagram for fplot_core::colormap:

## 5.4 fplot_list::container Type Reference

A container type allowing storage of most any Fortran type.

#### Public Member Functions

- procedure, public get => cntr_get_item
  *Retrieves the stored data.*
- procedure, public set => cntr_set_item
  *Stores the specified item in the container.*

#### Private Attributes

- class(∗), pointer m_data => null()
  *A pointer to a polymorphic variable allowing storage of any type.*

### 5.4.1 Detailed Description

A container type allowing storage of most any Fortran type.

Definition at line 27 of file fplot_list.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_list.f90

## 5.5 fplot_core::cool_colormap Type Reference

Defines a colormap consisting of "cool" colors.

Inheritance diagram for fplot_core::cool_colormap:

Collaboration diagram for fplot_core::cool_colormap:

#### Public Member Functions

- procedure, public get_color_string => ccm_get_clr
  *Gets the GNUPLOT string defining the color distribution.*

### 5.5.1 Detailed Description

Defines a colormap consisting of "cool" colors.

Definition at line 577 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.6 fplot_core::get_string_result Interface Reference

Retrieves a string from a plot_object.

#### Private Member Functions

- character(len=:) function, allocatable **get_string_result** (this)

### 5.6.1 Detailed Description

Retrieves a string from a plot_object.

#### Parameters

| | | |
|---|---|---|
| in | *this* | The plot_object object. |

#### Returns

The string.

Definition at line 958 of file fplot_core.f90.

The documentation for this interface was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.7 fplot_core::hot_colormap Type Reference

Defines a colormap consisting of "hot" colors.

Inheritance diagram for fplot_core::hot_colormap:

Collaboration diagram for fplot_core::hot_colormap:

#### Public Member Functions

- procedure, public get_color_string => hcm_get_clr
  *Gets the GNUPLOT string defining the color distribution.*

### 5.7.1 Detailed Description

Defines a colormap consisting of "hot" colors.

Definition at line 569 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.8 fplot_core::legend Type Reference

Defines a legend object.

Inheritance diagram for fplot_core::legend:

Collaboration diagram for fplot_core::legend:

#### Public Member Functions

- procedure, public get_draw_inside_axes => leg_get_inside

  *Gets a value determining if the legend should be drawn inside the axes border (true), or outside the axes border (false).*
- procedure, public set_draw_inside_axes => leg_set_inside

  *Sets a value determining if the legend should be drawn inside the axes border (true), or outside the axes border (false).*
- procedure, public get_draw_border => leg_get_box

  *Gets a value determining if the legend should have a border.*
- procedure, public set_draw_border => leg_set_box

  *Sets a value determining if the legend should have a border.*
- procedure, public get_horizontal_position => leg_get_horz_pos

  *Gets the horizontal position of the legend.*
- procedure, public set_horizontal_position => leg_set_horz_pos

  *Sets the horizontal position of the legend.*
- procedure, public get_vertical_position => leg_get_vert_pos

  *Gets the vertical position of the legend.*
- procedure, public set_vertical_position => leg_set_vert_pos

  *Gets the vertical position of the legend.*
- procedure, public get_is_visible => leg_get_visible

  *Gets a value determining if the legend is visible.*
- procedure, public set_is_visible => leg_set_visible

  *Sets a value determining if the legend is visible.*
- procedure, public get_command_string => leg_get_command_txt

  *Gets the command string defining the legend properties.*

#### Private Attributes

- logical m_inside = .true.

  *Legend on inside or outside of axes.*
- logical m_box = .true.

  *Draw a box around the legend.*
- character(len=20) m_horzposition = LEGEND_RIGHT

  *Defines the horizontal position.*
- character(len=20) m_vertposition = LEGEND_TOP

  *Defines the vertical position.*
- logical m_show = .true.

  *Determines if the legend is visible.*

### 5.8.1 Detailed Description

Defines a legend object.

Definition at line 426 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.9 fplot_list::list Type Reference

A generic list container.

Collaboration diagram for fplot_list::list:

### Public Member Functions

- procedure, public get_count => list_get_count
  *Gets the number of items in the list.*
- procedure, public get_capacity => list_get_capacity
  *Gets the capacity of the list.*
- procedure, public set_capacity => list_set_capacity
  *Sets the capacity of the list.*
- procedure, public get => list_get
  *Gets an item from the list.*
- procedure, public set => list_set
  *Sets an item into the list.*
- procedure, public push => list_push
  *Pushes an item onto the end of the list.*
- procedure, public pop => list_pop
  *Pops the last item from the end of the list.*
- procedure, public insert => list_insert
  *Inserts an item into the list.*
- procedure, public remove => list_remove
  *Removes an item from the list.*
- procedure, public clear => list_clear
  *Clears the contents of the list.*

### Private Attributes

- type(container), dimension(:), allocatable m_list
  *A collection of container objects.*
- integer(int32) m_count = 0
  *The actual number of items in m_list.*

#### 5.9.1 Detailed Description

A generic list container.

Definition at line 40 of file fplot_list.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_list.f90

## 5.10 fplot_core::pa_get_string_result Interface Reference

Retrieves a string from a plot_axis.

#### Private Member Functions

- character(len=:) function, allocatable **pa_get_string_result** (this)

### 5.10.1 Detailed Description

Retrieves a string from a plot_axis.

#### Parameters

| | | |
|---|---|---|
| in | *this* | The plot_axis object. |

#### Returns

The string.

Definition at line 988 of file fplot_core.f90.

The documentation for this interface was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.11 fplot_core::pd_get_string_result Interface Reference

Retrieves a string from a plot_data object.

#### Private Member Functions

- character(len=:) function, allocatable **pd_get_string_result** (this)

### 5.11.1 Detailed Description

Retrieves a string from a plot_data object.

#### Parameters

| | | |
|---|---|---|
| in | *this* | The plot_data object. |

#### Returns

The string.

Definition at line 978 of file fplot_core.f90.

The documentation for this interface was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.12 fplot_core::plot Type Reference

Defines the basic GNUPLOT plot.

Inheritance diagram for fplot_core::plot:

Collaboration diagram for fplot_core::plot:

**Public Member Functions**

- procedure, public free_resources => plt_clean_up
  *Cleans up resources held by the plot object.*
- procedure, public initialize => plt_init
  *Initializes the plot object.*
- procedure, public get_title => plt_get_title
  *Gets the plot's title.*
- procedure, public set_title => plt_set_title
  *Sets the plot's title.*
- procedure, public is_title_defined => plt_has_title
  *Gets a value determining if a title has been defined for the plot object.*
- procedure, public get_legend => plt_get_legend
  *Gets the plot's legend object.*
- procedure, public get_count => plt_get_count
  *Gets the number of stored plot_data objects.*
- procedure, public push => plt_push_data
  *Pushes a plot_data object onto the stack.*
- procedure, public pop => plt_pop_data
  *Pops the last plot_data object from the stack.*
- procedure, public clear_all => plt_clear_all
  *Removes all plot_data objects from the plot.*
- procedure, public get => plt_get
  *Gets a pointer to the requested plot_data object.*
- procedure, public set => plt_set
  *Sets the requested plot_data object into the plot.*
- procedure, public get_terminal => plt_get_term
  *Gets the GNUPLOT terminal object.*
- procedure, public get_show_gridlines => plt_get_show_grid
  *Gets a flag determining if the grid lines should be shown.*
- procedure, public set_show_gridlines => plt_set_show_grid
  *Sets a flag determining if the grid lines should be shown.*
- procedure, public draw => plt_draw
  *Launches GNUPLOT and draws the plot per the current state of the command list.*
- procedure, public save_file => plt_save
  *Saves a GNUPLOT command file.*
- procedure, public get_font_name => plt_get_font
  *Gets the name of the font used for plot text.*
- procedure, public set_font_name => plt_set_font
  *Sets the name of the font used for plot text.*
- procedure, public get_font_size => plt_get_font_size
  *Gets the size of the font used by the plot.*
- procedure, public set_font_size => plt_set_font_size
  *Sets the size of the font used by the plot.*
- procedure, public get_tics_inward => plt_get_tics_in
  *Gets a value determining if the axis tic marks should point inwards.*
- procedure, public set_tics_inward => plt_set_tics_in
  *Sets a value determining if the axis tic marks should point inwards.*
- procedure, public get_draw_border => plt_get_draw_border
  *Gets a value determining if the border should be drawn.*
- procedure, public set_draw_border => plt_set_draw_border
  *Sets a value determining if the border should be drawn.*

#### Private Attributes

- character(len=plotdata_max_name_length) m_title = ""
  *The plot title.*
- logical m_hastitle = .false.
  *Has a title?*
- class(terminal), pointer m_terminal => null()
  *The GNUPLOT terminal object to target.*
- type(list) m_data
  *A collection of plot_data items to plot.*
- type(legend), pointer m_legend => null()
  *The legend.*
- logical m_showgrid = .true.
  *Show grid lines?*
- logical m_ticsin = .true.
  *Point tic marks in?*
- logical m_drawborder = .true.
  *Draw the border?*

### 5.12.1 Detailed Description

Defines the basic GNUPLOT plot.

Definition at line 467 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.13 fplot_core::plot_2d Type Reference

A plot object defining a 2D plot.

Inheritance diagram for fplot_core::plot_2d:

Collaboration diagram for fplot_core::plot_2d:

#### Public Member Functions

- procedure, public initialize => p2d_init
  *Initializes the plot_2d object.*
- procedure, public get_command_string => p2d_get_cmd
  *Gets the GNUPLOT command string to represent this plot_2d object.*
- procedure, public get_x_axis => p2d_get_x_axis
  *Gets the x-axis object.*
- procedure, public get_y_axis => p2d_get_y_axis
  *Gets the y-axis object.*
- procedure, public get_y2_axis => p2d_get_y2_axis
  *Gets the secondary y-axis object.*
- procedure, public get_use_y2_axis => p2d_get_use_y2
  *Gets a flag determining if the secondary y-axis should be displayed.*
- procedure, public set_use_y2_axis => p2d_set_use_y2
  *Sets a flag determining if the secondary y-axis should be displayed.*

#### Private Member Functions

- final p2d_clean_up
  *Cleans up resources held by the plot_2d object.*

#### Private Attributes

- type(x_axis), pointer m_xaxis => null()
  *The x-axis.*
- type(y_axis), pointer m_yaxis => null()
  *The y-axis.*
- type(y2_axis), pointer m_y2axis => null()
  *The secondary y-axis.*
- logical m_usey2 = .false.
  *Display the secondary y axis?*

### 5.13.1 Detailed Description

A plot object defining a 2D plot.

Definition at line 782 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.14 fplot_core::plot_3d Type Reference

A plot object defining a 3D plot.

Inheritance diagram for fplot_core::plot_3d:

Collaboration diagram for fplot_core::plot_3d:

#### Public Member Functions

- procedure, public initialize => p3d_init
  *Initializes the plot_3d object.*
- procedure, public get_command_string => p3d_get_cmd
  *Gets the GNUPLOT command string to represent this plot_3d object.*
- procedure, public get_x_axis => p3d_get_x_axis
  *Gets the x-axis object.*
- procedure, public get_y_axis => p3d_get_y_axis
  *Gets the y-axis object.*
- procedure, public get_z_axis => p3d_get_z_axis
  *Gets the z-axis object.*
- procedure, public get_elevation => p3d_get_elevation
  *Gets the plot elevation angle.*
- procedure, public set_elevation => p3d_set_elevation
  *Sets the plot elevation angle.*
- procedure, public get_azimuth => p3d_get_azimuth
  *Gets the plot azimuth angle.*
- procedure, public set_azimuth => p3d_set_azimuth
  *Sets the plot azimuth angle.*
- procedure, public get_z_intersect_xy => p3d_get_z_axis_intersect
  *Gets a value determining if the z-axis should intersect the x-y plane.*
- procedure, public set_z_intersect_xy => p3d_set_z_axis_intersect
  *Sets a value determining if the z-axis should intersect the x-y plane.*

#### Private Member Functions

- final p3d_clean_up

  *Cleans up resources held by the plot_3d object.*

#### Private Attributes

- type(x_axis), pointer m_xaxis => null()

  *The x-axis.*
- type(y_axis), pointer m_yaxis => null()

  *The y-axis.*
- type(z_axis), pointer m_zaxis => null()

  *The z-axis.*
- real(real64) m_elevation = 60.0d0

  *The elevation angle.*
- real(real64) m_azimuth = 30.0d0

  *The azimuth.*
- logical m_zintersect = .true.

  *Z-axis intersect X-Y plane?*

### 5.14.1 Detailed Description

A plot object defining a 3D plot.

Definition at line 816 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.15 fplot_core::plot_axis Type Reference

Describes a single plot axis.

Inheritance diagram for fplot_core::plot_axis:

Collaboration diagram for fplot_core::plot_axis:

### Public Member Functions

- procedure, public get_title => pa_get_title

  *Gets the axis' title.*
- procedure, public set_title => pa_set_title

  *Sets the axis' title.*
- procedure, public is_title_defined => pa_has_title

  *Gets a value determining if a title has been defined for the plot_axis object.*
- procedure, public get_autoscale => pa_get_autoscale

  *Gets a logical value determining if the axis should be automatically scaled to fit the data.*
- procedure, public set_autoscale => pa_set_autoscale

  *Sets a logical value determining if the axis should be automatically scaled to fit the data.*
- procedure, public get_limits => pa_get_axis_limits

  *Gets the axis display limits, assuming autoscaling is not active for this axis.*
- procedure, public set_limits => pa_set_axis_limits

  *Sets the axis display limits, assuming autoscaling is not active for this axis.*
- procedure, public get_is_log_scaled => pa_get_log_scale

  *Gets a logical value defining if the axis should be log scaled.*
- procedure, public set_is_log_scaled => pa_set_log_scale

  *Sets a logical value defining if the axis should be log scaled.*
- procedure, public get_command_string => pa_get_cmd_string

  *Returns the appropriate GNUPLOT command string to define the plot_axis properties.*
- procedure, public get_zero_axis => pa_get_zero_axis

  *Gets a value determining if the axis should be drawn through zero of opposing axes.*
- procedure, public set_zero_axis => pa_set_zero_axis

  *Sets a value determining if the axis should be drawn through zero of opposing axes.*
- procedure, public get_zero_axis_line_width => pa_get_zero_axis_width

  *Gets the width of the line used to represent the zero axis line, if active.*
- procedure, public set_zero_axis_line_width => pa_set_zero_axis_width

  *Sets the width of the line used to represent the zero axis line, if active.*
- procedure(pa_get_string_result), deferred, public get_id_string

  *Gets a string identifying the axis as: x, y, z, y2, etc.*

### Private Attributes

- logical m_hastitle = .false.

  *Has a title.*
- character(len=plotdata_max_name_length) m_title = ""

  *The axis title.*
- logical m_autoscale = .true.

  *Autoscale?*
- real(real64), dimension(2) m_limits = [0.0d0, 1.0d0]

  *Display limits.*
- logical m_logscale = .false.

  *Log scaled?*
- logical m_zeroaxis = .false.

  *Zero axis?*
- real(real32) m_axiswidth = 1.0

  *The width, in pixels, of the zero axis line.*

#### 5.15.1 Detailed Description

Describes a single plot axis.

Definition at line 363 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.16 fplot_core::plot_data Type Reference

Provides a container for plot data.

Inheritance diagram for fplot_core::plot_data:

Collaboration diagram for fplot_core::plot_data:

#### Public Member Functions

- procedure, public get_name => pd_get_name
  *Gets the name to associate with this data set.*
- procedure, public set_name => pd_set_name
  *Sets the name to associate with this data set.*
- procedure(pd_get_string_result), deferred, public get_data_string
  *Gets the GNUPLOT command string containing the actual data to plot.*

#### Private Attributes

- character(len=plotdata_max_name_length) m_name = ""
  *The name of the data set.*

#### 5.16.1 Detailed Description

Provides a container for plot data.

Definition at line 347 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.17 fplot_core::plot_data_2d Type Reference

Defines a two-dimensional plot data set.

Inheritance diagram for fplot_core::plot_data_2d:

Collaboration diagram for fplot_core::plot_data_2d:

#### Public Member Functions

- procedure, public get_axes_string => pd2d_get_axes_cmd

  *Gets the GNUPLOT command string defining which axes the data is to be plotted against.*
- procedure, public get_data_string => pd2d_get_data_cmd

  *Gets the GNUPLOT command string containing the actual data to plot.*
- procedure, public get_count => pd2d_get_data_count

  *Gets the number of data points.*
- procedure, public get_x => pd2d_get_x_data

  *Gets the requested X data point.*
- procedure, public set_x => pd2d_set_x_data

  *Sets the requested X data point.*
- procedure, public get_y => pd2d_get_y_data

  *Gets the requested Y data point.*
- procedure, public set_y => pd2d_set_y_data

  *Sets the requested Y data point.*
- procedure, public get_draw_against_y2 => pd2d_get_draw_against_y2

  *Gets a value determining if the data should be plotted against the secondary y-axis.*
- procedure, public set_draw_against_y2 => pd2d_set_draw_against_y2

  *Sets a value determining if the data should be plotted against the secondary y-axis.*
- generic, public define_data => pd2d_set_data_1, pd2d_set_data_2

  *Defines the data set.*

#### Private Member Functions

- procedure **pd2d_set_data_1**
- procedure **pd2d_set_data_2**

#### Private Attributes

- real(real64), dimension(:,:), allocatable m_data

  *An N-by-2 matrix containing the x and y data points.*
- logical m_usey2 = .false.

  *Draw against the secondary y axis?*

### 5.17.1 Detailed Description

Defines a two-dimensional plot data set.

Definition at line 668 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.18 fplot_core::plot_data_3d Type Reference

Defines a three-dimensional plot data set.

Inheritance diagram for fplot_core::plot_data_3d:

Collaboration diagram for fplot_core::plot_data_3d:

### Public Member Functions

- procedure, public get_count => pd3d_get_data_count
  *Gets the number of data points.*
- procedure, public get_x => pd3d_get_x_data
  *Gets the requested X data point.*
- procedure, public set_x => pd3d_set_x_data
  *Sets the requested X data point.*
- procedure, public get_y => pd3d_get_y_data
  *Gets the requested Y data point.*
- procedure, public set_y => pd3d_set_y_data
  *Sets the requested Y data point.*
- procedure, public get_z => pd3d_get_z_data
  *Gets the requested Z data point.*
- procedure, public set_z => pd3d_set_z_data
  *Sets the requested Z data point.*
- procedure, public get_axes_string => pd3d_get_axes_cmd
  *Gets the GNUPLOT command string defining which axes the data is to be plotted against.*
- procedure, public get_data_string => pd3d_get_data_cmd
  *Gets the GNUPLOT command string containing the actual data to plot.*
- procedure, public define_data => pd3d_set_data_1
  *Defines the data set.*

### Private Attributes

- real(real64), dimension(:,:), allocatable m_data
  *An N-by-3 matrix containing the x, y, and z data points.*

### 5.18.1 Detailed Description

Defines a three-dimensional plot data set.

Definition at line 705 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.19 fplot_core::plot_object Type Reference

The base type for a GNUPLOT object.

Inheritance diagram for fplot_core::plot_object:

### Public Member Functions

- procedure(get_string_result), deferred, public get_command_string
  *Returns the appropriate GNUPLOT command string to define the plot object properties.*

### 5.19.1 Detailed Description

The base type for a GNUPLOT object.

Definition at line 230 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.20 fplot_core::png_terminal Type Reference

Defines a GNUPLOT PNG terminal object.

Inheritance diagram for fplot_core::png_terminal:

Collaboration diagram for fplot_core::png_terminal:

### Public Member Functions

- procedure, public get_filename => png_get_filename
  *Gets the filename for the output PNG file.*
- procedure, public set_filename => png_set_filename
  *Sets the filename for the output PNG file.*
- procedure, public get_id_string => png_get_term_string
  *Retrieves a GNUPLOT terminal identifier string.*
- procedure, public get_command_string => png_get_command_string
  *Returns the appropriate GNUPLOT command string to establish appropriate parameters.*

### Private Attributes

- character(len=3) m_id = "png"
  *The terminal ID string.*
- character(len=gnuplot_max_path_length) m_fname = "default.png"
  *The filename of the PNG file to write.*

#### 5.20.1 Detailed Description

Defines a GNUPLOT PNG terminal object.

Definition at line 327 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.21 fplot_core::qt_terminal Type Reference

Defines a GNUPLOT QT terminal object.

Inheritance diagram for fplot_core::qt_terminal:

Collaboration diagram for fplot_core::qt_terminal:

#### Public Member Functions

- procedure, public get_id_string => qt_get_term_string
  *Retrieves a GNUPLOT terminal identifier string.*

#### Private Attributes

- character(len=2) m_id = "qt"
  *The terminal ID string.*

#### 5.21.1 Detailed Description

Defines a GNUPLOT QT terminal object.

Definition at line 305 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.22 fplot_core::rainbow_colormap Type Reference

Defines a rainbow colormap.

Inheritance diagram for fplot_core::rainbow_colormap:

Collaboration diagram for fplot_core::rainbow_colormap:

#### Public Member Functions

- procedure, public get_color_string => rcm_get_clr
  *Gets the GNUPLOT string defining the color distribution.*

### 5.22.1 Detailed Description

Defines a rainbow colormap.

Definition at line 561 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.23 fplot_core::scatter_plot_data Type Reference

A plot_data object for describing scatter plot data sets.

Inheritance diagram for fplot_core::scatter_plot_data:

Collaboration diagram for fplot_core::scatter_plot_data:

#### Public Member Functions

- procedure, public get_command_string => spd_get_cmd
  *Gets the GNUPLOT command string to represent this scatter_plot_data object.*
- procedure, public get_line_width => spd_get_line_width
  *Gets the width of the line, in pixels.*
- procedure, public set_line_width => spd_set_line_width
  *Sets the width of the line, in pixels.*
- procedure, public get_line_style => spd_get_line_style
  *Gets the line style.*
- procedure, public set_line_style => spd_set_line_style
  *Sets the line style.*
- procedure, public get_line_color => spd_get_line_color
  *Gets the line color.*
- procedure, public set_line_color => spd_set_line_color
  *Sets the line color.*
- procedure, public get_draw_line => spd_get_draw_line
  *Gets a value determining if a line should be drawn.*
- procedure, public set_draw_line => spd_set_draw_line
  *Sets a value determining if a line should be drawn.*
- procedure, public get_draw_markers => spd_get_draw_markers
  *Gets a value determining if data point markers should be drawn.*
- procedure, public set_draw_markers => spd_set_draw_markers
  *Sets a value determining if data point markers should be drawn.*
- procedure, public get_marker_style => spd_get_marker_style
  *Gets the marker style.*

- procedure, public set_marker_style => spd_set_marker_style

  *Sets the marker style.*
- procedure, public get_marker_scaling => spd_get_marker_scaling

  *Gets the marker scaling.*
- procedure, public set_marker_scaling => spd_set_marker_scaling

  *Sets the marker scaling.*
- procedure, public get_marker_frequency => spd_get_marker_frequency

  *Gets the marker frequency.*
- procedure, public set_marker_frequency => spd_set_marker_frequency

  *Sets the marker frequency.*
- procedure, public get_use_auto_color => spd_get_use_auto_colors

  *Gets a value determining if GNUPLOT should automatically choose line colors.*
- procedure, public set_use_auto_color => spd_set_use_auto_colors

  *Sets a value determining if GNUPLOT should automatically choose line colors.*
- procedure(spd_get_int_value), deferred, public get_count

  *Gets the number of data points.*
- procedure(spd_get_value), deferred, public get_x

  *Gets the requested X data point.*
- procedure(spd_set_value), deferred, public set_x

  *Sets the requested X data point.*
- procedure(spd_get_value), deferred, public get_y

  *Gets the requested Y data point.*
- procedure(spd_set_value), deferred, public set_y

  *Sets the requested X data point.*
- procedure(spd_get_string_result), deferred, public get_axes_string

  *Gets the GNUPLOT command string defining which axes the data is to be plotted against.*

#### Private Attributes

- logical m_drawline = .true.

  *Draw the line?*
- logical m_drawmarkers = .false.

  *Draw the markers?*
- integer(int32) m_markerfrequency = 1

  *Marker frequency.*
- type(color) m_linecolor = CLR_BLUE

  *Line color.*
- real(real32) m_linewidth = 1.0

  *Line width.*
- integer(int32) m_linestyle = LINE_SOLID

  *Line style.*
- integer(int32) m_markertype = MARKER_X

  *Marker type.*
- real(real32) m_markersize = 1.0

  *Marker size multiplier.*
- logical m_useautocolor = .true.

  *Let GNUPLOT choose colors automatically.*

#### 5.23.1 Detailed Description

A plot_data object for describing scatter plot data sets.

Definition at line 587 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

### 5.24 fplot_core::spd_get_int_value Interface Reference

Retrieves an integer value from a scatter_plot_data object.

#### Private Member Functions

- pure integer(int32) function **spd_get_int_value** (this)

#### 5.24.1 Detailed Description

Retrieves an integer value from a scatter_plot_data object.

**Parameters**

| | | |
|---|---|---|
| in | *this* | The scatter_plot_data object. |

**Returns**

The requested value.

Definition at line 1024 of file fplot_core.f90.

The documentation for this interface was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

### 5.25 fplot_core::spd_get_string_result Interface Reference

Retrieves a string from a scatter_plot_data object.

#### Private Member Functions

- character(len=:) function, allocatable **spd_get_string_result** (this)

#### 5.25.1 Detailed Description

Retrieves a string from a scatter_plot_data object.

#### Parameters

| | | |
|---|---|---|
| in | *this* | The scatter_plot_data object. |

#### Returns

The string.

Definition at line 1035 of file fplot_core.f90.

The documentation for this interface was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.26 fplot_core::spd_get_value Interface Reference

Retrieves a numeric value from a scatter_plot_data object.

#### Private Member Functions

- pure real(real64) function **spd_get_value** (this, index)

### 5.26.1 Detailed Description

Retrieves a numeric value from a scatter_plot_data object.

#### Parameters

| | | |
|---|---|---|
| in | *this* | The scatter_plot_data object. |
| in | *index* | The index of the value to retrieve. |

#### Returns

The requested value.

Definition at line 999 of file fplot_core.f90.

The documentation for this interface was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.27 fplot_core::spd_set_value Interface Reference

Sets a numeric value into a scatter_plot_data object.

#### Private Member Functions

- subroutine **spd_set_value** (this, index, x)

### 5.27.1 Detailed Description

Sets a numeric value into a scatter_plot_data object.

#### Parameters

| | | |
|---|---|---|
| in,out | *this* | The scatter_plot_data object. |
| in | *index* | The index of the value to retrieve. |
| in | *x* | The value. |

Definition at line 1012 of file fplot_core.f90.

The documentation for this interface was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.28 fplot_core::surface_plot Type Reference

Inheritance diagram for fplot_core::surface_plot:

Collaboration diagram for fplot_core::surface_plot:

#### Public Member Functions

- procedure, public initialize => surf_init
  *Initializes the surface_plot object.*
- procedure, public get_show_hidden => surf_get_show_hidden
  *Gets a value indicating if hidden lines should be shown.*
- procedure, public set_show_hidden => surf_set_show_hidden
  *Sets a value indicating if hidden lines should be shown.*
- procedure, public get_command_string => surf_get_cmd
  *Gets the GNUPLOT command string to represent this plot_3d object.*
- procedure, public get_colormap => surf_get_colormap
  *Gets a pointer to the colormap object.*
- procedure, public set_colormap => surf_set_colormap
  *Sets the colormap object.*
- procedure, public get_allow_smoothing => surf_get_smooth
  *Gets a value determining if the plotted surfaces should be smoothed.*
- procedure, public set_allow_smoothing => surf_set_smooth
  *Sets a value determining if the plotted surfaces should be smoothed.*
- procedure, public get_show_contours => surf_get_show_contours
  *Gets a value determining if a contour plot should be drawn in conjunction with the surface plot.*
- procedure, public set_show_contours => surf_set_show_contours
  *Sets a value determining if a contour plot should be drawn in conjunction with the surface plot.*
- procedure, public get_show_colorbar => surf_get_show_colorbar
  *Gets a value determining if the colorbar should be shown.*
- procedure, public set_show_colorbar => surf_set_show_colorbar
  *Sets a value determining if the colorbar should be shown.*

#### Private Member Functions

- final surf_clean_up

  *Cleans up resources held by the surface_plot object.*

#### Private Attributes

- logical m_showhidden = .false.

  *Show hidden lines.*
- class(colormap), pointer m_colormap

  *The colormap.*
- logical m_smooth = .true.

  *Smooth the surface?*
- logical m_contour = .false.

  *Show a contour plot as well as the surface plot?*
- logical m_showcolorbar = .true.

  *Show the colorbar?*

### 5.28.1 Detailed Description

Definition at line 862 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.29 fplot_core::surface_plot_data Type Reference

Provides a three-dimensional surface plot data set.

Inheritance diagram for fplot_core::surface_plot_data:

Collaboration diagram for fplot_core::surface_plot_data:

#### Public Member Functions

- procedure, public get_size => surfd_get_size

  *Gets the size of the stored data set.*
- procedure, public get_x => surfd_get_x

  *Gets the requested X data point.*
- procedure, public set_x => surfd_set_x

  *Sets the requested X data point.*
- procedure, public get_y => surfd_get_y

  *Gets the requested Y data point.*
- procedure, public set_y => surfd_set_y

  *Sets the requested Y data point.*
- procedure, public get_z => surfd_get_z

  *Gets the requested Z data point.*

- procedure, public set_z => surfd_set_z

  *Sets the requested Z data point.*
- procedure, public get_use_wireframe => surfd_get_wireframe

  *Gets a value determining if a wireframe mesh should be displayed.*
- procedure, public set_use_wireframe => surfd_set_wireframe

  *Sets a value determining if a wireframe mesh should be displayed.*
- procedure, public get_command_string => surfd_get_cmd

  *Gets the GNUPLOT command string to represent this surface_plot_data object.*
- procedure, public get_data_string => surfd_get_data_cmd

  *Gets the GNUPLOT command string containing the actual data to plot.*
- procedure, public define_data => surfd_set_data_1

  *Defines the data set.*

#### Private Attributes

- real(real64), dimension(:,:), allocatable m_x

  *Stores the x-coordinate data.*
- real(real64), dimension(:,:), allocatable m_y

  *Stores the y-coordinate data.*
- real(real64), dimension(:,:), allocatable m_z

  *Stores the z-coordinate data.*
- logical m_wireframe = .false.

  *Set to true to display a wireframe of the surface; else, just a smooth surface will be drawn.*

### 5.29.1 Detailed Description

Provides a three-dimensional surface plot data set.

Definition at line 736 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.30 fplot_core::term_get_string_result Interface Reference

Retrieves a string from a terminal.

#### Private Member Functions

- character(len=:) function, allocatable **term_get_string_result** (this)

### 5.30.1 Detailed Description

Retrieves a string from a terminal.

#### Parameters

| | | |
|---|---|---|
| in | *this* | The terminal object. |

#### Returns

The string.

Definition at line 968 of file fplot_core.f90.

The documentation for this interface was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.31 fplot_core::terminal Type Reference

Defines a GNUPLOT terminal object.

Inheritance diagram for fplot_core::terminal:

Collaboration diagram for fplot_core::terminal:

#### Public Member Functions

- procedure, public get_window_width => term_get_window_width
  *Gets the width of the plot window.*
- procedure, public set_window_width => term_set_window_width
  *Sets the width of the plot window.*
- procedure, public get_window_height => term_get_window_height
  *Gets the height of the plot window.*
- procedure, public set_window_height => term_set_window_height
  *Sets the height of the plot window.*
- procedure, public get_command_string => term_get_command_string
  *Returns the appropriate GNUPLOT command string to establish appropriate parameters.*
- procedure, public get_plot_window_number =>term_get_plot_window_number
  *Gets the targeted plot window number.*
- procedure, public set_plot_window_number =>term_set_plot_window_number
  *Sets the targeted plot window number.*
- procedure, public get_title => term_get_title
  *Gets the plot window's title.*
- procedure, public set_title => term_set_title
  *Sets the plot window's title.*
- procedure, public get_font_name => term_get_font_name
  *Gets the name of the font used for text displayed by the graph.*
- procedure, public set_font_name => term_set_font_name
  *Sets the name of the font used for text displayed by the graph.*
- procedure, public get_font_size => term_get_font_size
  *Gets the size of the font used by the graph.*
- procedure, public set_font_size => term_set_font_size
  *Sets the size of the font used by the graph.*
- procedure(term_get_string_result), deferred, public get_id_string
  *Gets the GNUPLOT terminal identification string.*

#### Private Attributes

- integer(int32) m_windowheight = GNUPLOT_DEFAULT_WINDOW_HEIGHT

  *The window height, in pixels.*
- integer(int32) m_windowwidth = GNUPLOT_DEFAULT_WINDOW_WIDTH

  *The window width, in pixels.*
- integer(int32) m_termid = 0

  *The plot window number.*
- character(len=gnuplot_max_label_length) m_title = ""

  *The plot window title.*
- logical m_hastitle = .false.

  *Determines if a plot title is defined.*
- character(len=gnuplot_max_label_length) m_fontname = GNUPLOT_DEFAULT_FONTNAME

  *The font used by the graph.*
- integer(int32) m_fontsize = GNUPLOT_DEFAULT_FONT_SIZE

  *The size of the font used by the graph.*

### 5.31.1 Detailed Description

Defines a GNUPLOT terminal object.

Definition at line 239 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.32 fplot_core::windows_terminal Type Reference

Defines a GNUPLOT Win32 terminal object.

Inheritance diagram for fplot_core::windows_terminal:

Collaboration diagram for fplot_core::windows_terminal:

#### Public Member Functions

- procedure, public get_id_string => wt_get_term_string

  *Retrieves a GNUPLOT terminal identifier string.*

#### Private Attributes

- character(len=3) m_id = "win"

  *The terminal ID string.*

#### 5.32.1 Detailed Description

Defines a GNUPLOT Win32 terminal object.

Definition at line 294 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.33 fplot_core::wxt_terminal Type Reference

Defines a GNUPLOT WXT terminal object.

Inheritance diagram for fplot_core::wxt_terminal:

Collaboration diagram for fplot_core::wxt_terminal:

#### Public Member Functions

- procedure, public get_id_string => wxt_get_term_string
  *Retrieves a GNUPLOT terminal identifier string.*

#### Private Attributes

- character(len=3) m_id = "wxt"
  *The terminal ID string.*

#### 5.33.1 Detailed Description

Defines a GNUPLOT WXT terminal object.

Definition at line 316 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.34 fplot_core::x_axis Type Reference

An x-axis object.

Inheritance diagram for fplot_core::x_axis:

Collaboration diagram for fplot_core::x_axis:

#### Public Member Functions

- procedure, public get_id_string => xa_get_id
  *Gets the axis identification string.*

#### Private Attributes

- character m_id = "x"
  *The ID character.*

### 5.34.1 Detailed Description

An x-axis object.

Definition at line 912 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.35 fplot_core::y2_axis Type Reference

A secondary y-axis object.

Inheritance diagram for fplot_core::y2_axis:

Collaboration diagram for fplot_core::y2_axis:

#### Public Member Functions

- procedure, public get_id_string => y2a_get_id
  *Gets the axis identification string.*

#### Private Attributes

- character(len=2) m_id = "y2"
  *The ID character.*

### 5.35.1 Detailed Description

A secondary y-axis object.

Definition at line 932 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.36 fplot_core::y_axis Type Reference

A y-axis object.

Inheritance diagram for fplot_core::y_axis:

Collaboration diagram for fplot_core::y_axis:

#### Public Member Functions

- procedure, public get_id_string => ya_get_id
  *Gets the axis identification string.*

#### Private Attributes

- character m_id = "y"
  *The ID character.*

### 5.36.1 Detailed Description

A y-axis object.

Definition at line 922 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

## 5.37 fplot_core::z_axis Type Reference

A z-axis object.

Inheritance diagram for fplot_core::z_axis:

Collaboration diagram for fplot_core::z_axis:

#### Public Member Functions

- procedure, public get_id_string => za_get_id
  *Gets the axis identification string.*

#### Private Attributes

- character m_id = "z"
  *The ID character.*

### 5.37.1 Detailed Description

A z-axis object.

Definition at line 942 of file fplot_core.f90.

The documentation for this type was generated from the following file:

- /home/jason/Documents/Code/fplot/src/fplot_core.f90

# Index


ccm_get_clr
    fplot_core, 16
clr_copy_from
    fplot_core, 16
clr_to_hex_string
    fplot_core, 16
cm_get_cmd
    fplot_core, 16
cntr_get_item
    fplot_list, 66
cntr_set_item
    fplot_list, 67

fplot_core, 5
    ccm_get_clr, 16
    clr_copy_from, 16
    clr_to_hex_string, 16
    cm_get_cmd, 16
    hcm_get_clr, 17
    leg_get_box, 17
    leg_get_command_txt, 17
    leg_get_horz_pos, 17
    leg_get_inside, 18
    leg_get_vert_pos, 18
    leg_get_visible, 18
    leg_set_box, 19
    leg_set_horz_pos, 19
    leg_set_inside, 19
    leg_set_vert_pos, 19
    leg_set_visible, 20
    p2d_clean_up, 20
    p2d_get_cmd, 20
    p2d_get_use_y2, 20
    p2d_get_x_axis, 21
    p2d_get_y2_axis, 21
    p2d_get_y_axis, 21
    p2d_init, 22
    p2d_set_use_y2, 22
    p3d_clean_up, 22
    p3d_get_azimuth, 23
    p3d_get_cmd, 23
    p3d_get_elevation, 23
    p3d_get_x_axis, 23
    p3d_get_y_axis, 24
    p3d_get_z_axis, 24
    p3d_get_z_axis_intersect, 24
    p3d_init, 25
    p3d_set_azimuth, 25
    p3d_set_elevation, 25
    p3d_set_z_axis_intersect, 26
    pa_get_autoscale, 26
    pa_get_axis_limits, 26
    pa_get_cmd_string, 26
    pa_get_log_scale, 27
    pa_get_title, 27
    pa_get_zero_axis, 27
    pa_get_zero_axis_width, 28
    pa_has_title, 28
    pa_set_autoscale, 28
    pa_set_axis_limits, 28
    pa_set_log_scale, 29
    pa_set_title, 29
    pa_set_zero_axis, 29
    pa_set_zero_axis_width, 29
    pd2d_get_axes_cmd, 30
    pd2d_get_data_cmd, 30
    pd2d_get_data_count, 30
    pd2d_get_draw_against_y2, 31
    pd2d_get_x_data, 31
    pd2d_get_y_data, 31
    pd2d_set_data_1, 32
    pd2d_set_data_2, 32
    pd2d_set_draw_against_y2, 32
    pd2d_set_x_data, 33
    pd2d_set_y_data, 33
    pd3d_get_axes_cmd, 33
    pd3d_get_data_cmd, 33
    pd3d_get_data_count, 34
    pd3d_get_x_data, 34
    pd3d_get_y_data, 34
    pd3d_get_z_data, 35
    pd3d_set_data_1, 35
    pd3d_set_x_data, 35
    pd3d_set_y_data, 36
    pd3d_set_z_data, 36
    pd_get_name, 36
    pd_set_name, 37
    plt_clean_up, 37
    plt_clear_all, 37
    plt_draw, 37
    plt_get, 38
    plt_get_count, 38
    plt_get_draw_border, 38
    plt_get_font, 39
    plt_get_font_size, 39
    plt_get_legend, 39
    plt_get_show_grid, 39
    plt_get_term, 40
    plt_get_tics_in, 40
    plt_get_title, 40
    plt_has_title, 41
    plt_init, 41
    plt_pop_data, 41
    plt_push_data, 42
    plt_save, 42
    plt_set, 42
    plt_set_draw_border, 43
    plt_set_font, 43
    plt_set_font_size, 43
    plt_set_show_grid, 43

plt_set_tics_in, 43
plt_set_title, 44
png_get_command_string, 44
png_get_filename, 44
png_get_term_string, 44
png_set_filename, 45
qt_get_term_string, 45
rcm_get_clr, 45
spd_get_cmd, 46
spd_get_draw_line, 46
spd_get_draw_markers, 46
spd_get_line_color, 46
spd_get_line_style, 47
spd_get_line_width, 47
spd_get_marker_frequency, 47
spd_get_marker_scaling, 48
spd_get_marker_style, 48
spd_get_use_auto_colors, 49
spd_set_draw_line, 49
spd_set_draw_markers, 49
spd_set_line_color, 49
spd_set_line_style, 50
spd_set_line_width, 50
spd_set_marker_frequency, 50
spd_set_marker_scaling, 51
spd_set_marker_style, 51
spd_set_use_auto_colors, 52
surf_clean_up, 52
surf_get_cmd, 52
surf_get_colormap, 52
surf_get_show_colorbar, 53
surf_get_show_contours, 53
surf_get_show_hidden, 53
surf_get_smooth, 53
surf_init, 54
surf_set_colormap, 54
surf_set_show_colorbar, 55
surf_set_show_contours, 55
surf_set_show_hidden, 55
surf_set_smooth, 55
surfd_get_cmd, 55
surfd_get_data_cmd, 56
surfd_get_size, 56
surfd_get_wireframe, 56
surfd_get_x, 57
surfd_get_y, 57
surfd_get_z, 57
surfd_set_data_1, 58
surfd_set_wireframe, 58
surfd_set_x, 58
surfd_set_y, 59
surfd_set_z, 59
term_get_command_string, 59
term_get_font_name, 60
term_get_font_size, 60
term_get_plot_window_number, 60
term_get_title, 60
term_get_window_height, 61
term_get_window_width, 61
term_set_font_name, 61
term_set_font_size, 62
term_set_plot_window_number, 62
term_set_title, 62
term_set_window_height, 62
term_set_window_width, 63
wt_get_term_string, 63
wxt_get_term_string, 63
xa_get_id, 63
y2a_get_id, 64
ya_get_id, 64
za_get_id, 64
fplot_core::cm_get_string_result, 70
fplot_core::color, 71
fplot_core::colormap, 71
fplot_core::cool_colormap, 72
fplot_core::get_string_result, 72
fplot_core::hot_colormap, 73
fplot_core::legend, 73
fplot_core::pa_get_string_result, 75
fplot_core::pd_get_string_result, 77
fplot_core::plot, 77
fplot_core::plot_2d, 79
fplot_core::plot_3d, 80
fplot_core::plot_axis, 81
fplot_core::plot_data, 83
fplot_core::plot_data_2d, 83
fplot_core::plot_data_3d, 85
fplot_core::plot_object, 86
fplot_core::png_terminal, 86
fplot_core::qt_terminal, 87
fplot_core::rainbow_colormap, 87
fplot_core::scatter_plot_data, 88
fplot_core::spd_get_int_value, 90
fplot_core::spd_get_string_result, 90
fplot_core::spd_get_value, 91
fplot_core::spd_set_value, 91
fplot_core::surface_plot, 92
fplot_core::surface_plot_data, 93
fplot_core::term_get_string_result, 94
fplot_core::terminal, 95
fplot_core::windows_terminal, 96
fplot_core::wxt_terminal, 97
fplot_core::x_axis, 97
fplot_core::y2_axis, 98
fplot_core::y_axis, 99
fplot_core::z_axis, 99
fplot_errors, 65
fplot_list, 65
cntr_get_item, 66
cntr_set_item, 67
list_clear, 67
list_get, 67
list_get_capacity, 67
list_get_count, 68
list_insert, 68
list_pop, 68

list_push, 68
list_remove, 69
list_set, 69
list_set_capacity, 69
fplot_list::container, 71
fplot_list::list, 75

hcm_get_clr
fplot_core, 17

leg_get_box
fplot_core, 17
leg_get_command_txt
fplot_core, 17
leg_get_horz_pos
fplot_core, 17
leg_get_inside
fplot_core, 18
leg_get_vert_pos
fplot_core, 18
leg_get_visible
fplot_core, 18
leg_set_box
fplot_core, 19
leg_set_horz_pos
fplot_core, 19
leg_set_inside
fplot_core, 19
leg_set_vert_pos
fplot_core, 19
leg_set_visible
fplot_core, 20
list_clear
fplot_list, 67
list_get
fplot_list, 67
list_get_capacity
fplot_list, 67
list_get_count
fplot_list, 68
list_insert
fplot_list, 68
list_pop
fplot_list, 68
list_push
fplot_list, 68
list_remove
fplot_list, 69
list_set
fplot_list, 69
list_set_capacity
fplot_list, 69

p2d_clean_up
fplot_core, 20
p2d_get_cmd
fplot_core, 20
p2d_get_use_y2
fplot_core, 20
p2d_get_x_axis
fplot_core, 21
p2d_get_y2_axis
fplot_core, 21
p2d_get_y_axis
fplot_core, 21
p2d_init
fplot_core, 22
p2d_set_use_y2
fplot_core, 22
p3d_clean_up
fplot_core, 22
p3d_get_azimuth
fplot_core, 23
p3d_get_cmd
fplot_core, 23
p3d_get_elevation
fplot_core, 23
p3d_get_x_axis
fplot_core, 23
p3d_get_y_axis
fplot_core, 24
p3d_get_z_axis
fplot_core, 24
p3d_get_z_axis_intersect
fplot_core, 24
p3d_init
fplot_core, 25
p3d_set_azimuth
fplot_core, 25
p3d_set_elevation
fplot_core, 25
p3d_set_z_axis_intersect
fplot_core, 26
pa_get_autoscale
fplot_core, 26
pa_get_axis_limits
fplot_core, 26
pa_get_cmd_string
fplot_core, 26
pa_get_log_scale
fplot_core, 27
pa_get_title
fplot_core, 27
pa_get_zero_axis
fplot_core, 27
pa_get_zero_axis_width
fplot_core, 28
pa_has_title
fplot_core, 28
pa_set_autoscale
fplot_core, 28
pa_set_axis_limits
fplot_core, 28
pa_set_log_scale
fplot_core, 29
pa_set_title
fplot_core, 29

pa_set_zero_axis
fplot_core, 29
pa_set_zero_axis_width
fplot_core, 29
pd2d_get_axes_cmd
fplot_core, 30
pd2d_get_data_cmd
fplot_core, 30
pd2d_get_data_count
fplot_core, 30
pd2d_get_draw_against_y2
fplot_core, 31
pd2d_get_x_data
fplot_core, 31
pd2d_get_y_data
fplot_core, 31
pd2d_set_data_1
fplot_core, 32
pd2d_set_data_2
fplot_core, 32
pd2d_set_draw_against_y2
fplot_core, 32
pd2d_set_x_data
fplot_core, 33
pd2d_set_y_data
fplot_core, 33
pd3d_get_axes_cmd
fplot_core, 33
pd3d_get_data_cmd
fplot_core, 33
pd3d_get_data_count
fplot_core, 34
pd3d_get_x_data
fplot_core, 34
pd3d_get_y_data
fplot_core, 34
pd3d_get_z_data
fplot_core, 35
pd3d_set_data_1
fplot_core, 35
pd3d_set_x_data
fplot_core, 35
pd3d_set_y_data
fplot_core, 36
pd3d_set_z_data
fplot_core, 36
pd_get_name
fplot_core, 36
pd_set_name
fplot_core, 37
plt_clean_up
fplot_core, 37
plt_clear_all
fplot_core, 37
plt_draw
fplot_core, 37
plt_get
fplot_core, 38
plt_get_count
fplot_core, 38
plt_get_draw_border
fplot_core, 38
plt_get_font
fplot_core, 39
plt_get_font_size
fplot_core, 39
plt_get_legend
fplot_core, 39
plt_get_show_grid
fplot_core, 39
plt_get_term
fplot_core, 40
plt_get_tics_in
fplot_core, 40
plt_get_title
fplot_core, 40
plt_has_title
fplot_core, 41
plt_init
fplot_core, 41
plt_pop_data
fplot_core, 41
plt_push_data
fplot_core, 42
plt_save
fplot_core, 42
plt_set
fplot_core, 42
plt_set_draw_border
fplot_core, 43
plt_set_font
fplot_core, 43
plt_set_font_size
fplot_core, 43
plt_set_show_grid
fplot_core, 43
plt_set_tics_in
fplot_core, 43
plt_set_title
fplot_core, 44
png_get_command_string
fplot_core, 44
png_get_filename
fplot_core, 44
png_get_term_string
fplot_core, 44
png_set_filename
fplot_core, 45

qt_get_term_string
fplot_core, 45

rcm_get_clr
fplot_core, 45

spd_get_cmd
fplot_core, 46

spd_get_draw_line
fplot_core, 46
spd_get_draw_markers
fplot_core, 46
spd_get_line_color
fplot_core, 46
spd_get_line_style
fplot_core, 47
spd_get_line_width
fplot_core, 47
spd_get_marker_frequency
fplot_core, 47
spd_get_marker_scaling
fplot_core, 48
spd_get_marker_style
fplot_core, 48
spd_get_use_auto_colors
fplot_core, 49
spd_set_draw_line
fplot_core, 49
spd_set_draw_markers
fplot_core, 49
spd_set_line_color
fplot_core, 49
spd_set_line_style
fplot_core, 50
spd_set_line_width
fplot_core, 50
spd_set_marker_frequency
fplot_core, 50
spd_set_marker_scaling
fplot_core, 51
spd_set_marker_style
fplot_core, 51
spd_set_use_auto_colors
fplot_core, 52
surf_clean_up
fplot_core, 52
surf_get_cmd
fplot_core, 52
surf_get_colormap
fplot_core, 52
surf_get_show_colorbar
fplot_core, 53
surf_get_show_contours
fplot_core, 53
surf_get_show_hidden
fplot_core, 53
surf_get_smooth
fplot_core, 53
surf_init
fplot_core, 54
surf_set_colormap
fplot_core, 54
surf_set_show_colorbar
fplot_core, 55
surf_set_show_contours
fplot_core, 55
surf_set_show_hidden
fplot_core, 55
surf_set_smooth
fplot_core, 55
surfd_get_cmd
fplot_core, 55
surfd_get_data_cmd
fplot_core, 56
surfd_get_size
fplot_core, 56
surfd_get_wireframe
fplot_core, 56
surfd_get_x
fplot_core, 57
surfd_get_y
fplot_core, 57
surfd_get_z
fplot_core, 57
surfd_set_data_1
fplot_core, 58
surfd_set_wireframe
fplot_core, 58
surfd_set_x
fplot_core, 58
surfd_set_y
fplot_core, 59
surfd_set_z
fplot_core, 59

term_get_command_string
fplot_core, 59
term_get_font_name
fplot_core, 60
term_get_font_size
fplot_core, 60
term_get_plot_window_number
fplot_core, 60
term_get_title
fplot_core, 60
term_get_window_height
fplot_core, 61
term_get_window_width
fplot_core, 61
term_set_font_name
fplot_core, 61
term_set_font_size
fplot_core, 62
term_set_plot_window_number
fplot_core, 62
term_set_title
fplot_core, 62
term_set_window_height
fplot_core, 62
term_set_window_width
fplot_core, 63

wt_get_term_string
fplot_core, 63
wxt_get_term_string

fplot_core, 63

xa_get_id
fplot_core, 63

y2a_get_id
fplot_core, 64
ya_get_id
fplot_core, 64

za_get_id
fplot_core, 64