

fplot

1.1.0

Generated by Doxygen 1.8.11

## Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Modules Index</b>   | <b>1</b>  |
| 1.1      | Modules List . . . . .   | 1         |
| <b>2</b> | <b>Data Type Index</b>   | <b>2</b>  |
| 2.1      | Class Hierarchy . . . . .                                      | 2         |
| <b>3</b> | <b>Data Type Index</b>   | <b>3</b>  |
| 3.1      | Data Types List . . . . .                                      | 3         |
| <b>4</b> | <b>Module Documentation</b>                                    | <b>5</b>  |
| 4.1      | fplot_core Module Reference . . . . .                          | 5         |
| 4.1.1    | Detailed Description . . . . .                                 | 15        |
| 4.1.2    | Function/Subroutine Documentation . . . . .                    | 15        |
| 4.2      | fplot_errors Module Reference . . . . .                        | 65        |
| 4.2.1    | Detailed Description . . . . .                                 | 66        |
| <b>5</b> | <b>Data Type Documentation</b>                                 | <b>66</b> |
| 5.1      | fplot_core::cm_get_string_result Interface Reference . . . . . | 66        |
| 5.1.1    | Detailed Description . . . . .                                 | 66        |
| 5.2      | fplot_core::color Type Reference . . . . .                     | 67        |
| 5.2.1    | Detailed Description . . . . .                                 | 67        |
| 5.3      | fplot_core::colormap Type Reference . . . . .                  | 67        |
| 5.4      | fplot_core::cool_colormap Type Reference . . . . .             | 67        |
| 5.4.1    | Detailed Description . . . . .                                 | 68        |
| 5.5      | fplot_core::get_string_result Interface Reference . . . . .    | 68        |
| 5.5.1    | Detailed Description . . . . .                                 | 68        |
| 5.6      | fplot_core::hot_colormap Type Reference . . . . .              | 68        |
| 5.6.1    | Detailed Description . . . . .                                 | 69        |
| 5.7      | fplot_core::legend Type Reference . . . . .                    | 69        |
| 5.7.1    | Detailed Description . . . . .                                 | 70        |
| 5.8      | fplot_core::pa_get_string_result Interface Reference . . . . . | 70        |

|  |    |
|--|----|
| 5.8.1 Detailed Description . . . . .                                 | 70 |
| 5.9 fplot_core::pd_get_string_result Interface Reference . . . . .   | 71 |
| 5.9.1 Detailed Description . . . . .                                 | 71 |
| 5.10 fplot_core::plot Type Reference . . . . .                       | 71 |
| 5.10.1 Detailed Description . . . . .                                | 73 |
| 5.11 fplot_core::plot_2d Type Reference . . . . .                    | 73 |
| 5.11.1 Detailed Description . . . . .                                | 74 |
| 5.12 fplot_core::plot_3d Type Reference . . . . .                    | 74 |
| 5.12.1 Detailed Description . . . . .                                | 75 |
| 5.13 fplot_core::plot_axis Type Reference . . . . .                  | 75 |
| 5.13.1 Detailed Description . . . . .                                | 77 |
| 5.14 fplot_core::plot_data Type Reference . . . . .                  | 77 |
| 5.14.1 Detailed Description . . . . .                                | 77 |
| 5.15 fplot_core::plot_data_2d Type Reference . . . . .               | 77 |
| 5.15.1 Detailed Description . . . . .                                | 78 |
| 5.16 fplot_core::plot_data_3d Type Reference . . . . .               | 79 |
| 5.16.1 Detailed Description . . . . .                                | 79 |
| 5.17 fplot_core::plot_object Type Reference . . . . .                | 80 |
| 5.17.1 Detailed Description . . . . .                                | 80 |
| 5.18 fplot_core::png_terminal Type Reference . . . . .               | 80 |
| 5.18.1 Detailed Description . . . . .                                | 81 |
| 5.19 fplot_core::qt_terminal Type Reference . . . . .                | 81 |
| 5.19.1 Detailed Description . . . . .                                | 81 |
| 5.20 fplot_core::rainbow_colormap Type Reference . . . . .           | 81 |
| 5.20.1 Detailed Description . . . . .                                | 82 |
| 5.21 fplot_core::scatter_plot_data Type Reference . . . . .          | 82 |
| 5.21.1 Detailed Description . . . . .                                | 84 |
| 5.22 fplot_core::spd_get_int_value Interface Reference . . . . .     | 84 |
| 5.22.1 Detailed Description . . . . .                                | 84 |
| 5.23 fplot_core::spd_get_string_result Interface Reference . . . . . | 84 |

|   |           |
|---|-----------|
| 5.23.1 Detailed Description . . . . .                                 | 84        |
| 5.24 fplot_core::spd_get_value Interface Reference . . . . .          | 85        |
| 5.24.1 Detailed Description . . . . .                                 | 85        |
| 5.25 fplot_core::spd_set_value Interface Reference . . . . .          | 85        |
| 5.25.1 Detailed Description . . . . .                                 | 86        |
| 5.26 fplot_core::surface_plot Type Reference . . . . .                | 86        |
| 5.26.1 Detailed Description . . . . .                                 | 87        |
| 5.27 fplot_core::surface_plot_data Type Reference . . . . .           | 87        |
| 5.27.1 Detailed Description . . . . .                                 | 88        |
| 5.28 fplot_core::term_get_string_result Interface Reference . . . . . | 89        |
| 5.28.1 Detailed Description . . . . .                                 | 89        |
| 5.29 fplot_core::terminal Type Reference . . . . .                    | 89        |
| 5.29.1 Detailed Description . . . . .                                 | 90        |
| 5.30 fplot_core::windows_terminal Type Reference . . . . .            | 90        |
| 5.30.1 Detailed Description . . . . .                                 | 91        |
| 5.31 fplot_core::wxt_terminal Type Reference . . . . .                | 91        |
| 5.31.1 Detailed Description . . . . .                                 | 91        |
| 5.32 fplot_core::x_axis Type Reference . . . . .                      | 92        |
| 5.32.1 Detailed Description . . . . .                                 | 92        |
| 5.33 fplot_core::y2_axis Type Reference . . . . .                     | 92        |
| 5.33.1 Detailed Description . . . . .                                 | 93        |
| 5.34 fplot_core::y_axis Type Reference . . . . .                      | 93        |
| 5.34.1 Detailed Description . . . . .                                 | 93        |
| 5.35 fplot_core::z_axis Type Reference . . . . .                      | 93        |
| 5.35.1 Detailed Description . . . . .                                 | 94        |
| <b>Index</b>  | <b>95</b> |

## 1 Modules Index

### 1.1 Modules List

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

|                              |    |
|------------------------------|----|
| <a href="#">fplot_core</a>   |    |
| <a href="#">fplot_core</a>   | 5  |
| <a href="#">fplot_errors</a> |    |
| <a href="#">plot_errors</a>  | 65 |

## 2 Data Type Index

### 2.1 Class Hierarchy

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

|  |    |
|--|----|
| <a href="#">fplot_core::cm_get_string_result</a> | 66 |
| <a href="#">fplot_core::color</a>                | 67 |
| <a href="#">fplot_core::get_string_result</a>    | 68 |
| <a href="#">fplot_core::pa_get_string_result</a> | 70 |
| <a href="#">fplot_core::pd_get_string_result</a> | 71 |
| <a href="#">fplot_core::plot_object</a>          | 80 |
| <a href="#">fplot_core::colormap</a>             | 67 |
| <a href="#">fplot_core::cool_colormap</a>        | 67 |
| <a href="#">fplot_core::hot_colormap</a>         | 68 |
| <a href="#">fplot_core::rainbow_colormap</a>     | 81 |
| <a href="#">fplot_core::legend</a>               | 69 |
| <a href="#">fplot_core::plot</a>                 | 71 |
| <a href="#">fplot_core::plot_2d</a>              | 73 |
| <a href="#">fplot_core::plot_3d</a>              | 74 |
| <a href="#">fplot_core::surface_plot</a>         | 86 |
| <a href="#">fplot_core::plot_axis</a>            | 75 |
| <a href="#">fplot_core::x_axis</a>               | 92 |
| <a href="#">fplot_core::y2_axis</a>              | 92 |
| <a href="#">fplot_core::y_axis</a>               | 93 |
| <a href="#">fplot_core::z_axis</a>               | 93 |
| <a href="#">fplot_core::plot_data</a>            | 77 |
| <a href="#">fplot_core::scatter_plot_data</a>    | 82 |
| <a href="#">fplot_core::plot_data_2d</a>         | 77 |
| <a href="#">fplot_core::plot_data_3d</a>         | 79 |

|  |    |
|--|----|
| <a href="#">fplot_core::surface_plot_data</a>      | 87 |
| <a href="#">fplot_core::terminal</a>               | 89 |
| <a href="#">fplot_core::png_terminal</a>           | 80 |
| <a href="#">fplot_core::qt_terminal</a>            | 81 |
| <a href="#">fplot_core::windows_terminal</a>       | 90 |
| <a href="#">fplot_core::wxt_terminal</a>           | 91 |
| <a href="#">fplot_core::spd_get_int_value</a>      | 84 |
| <a href="#">fplot_core::spd_get_string_result</a>  | 84 |
| <a href="#">fplot_core::spd_get_value</a>          | 85 |
| <a href="#">fplot_core::spd_set_value</a>          | 85 |
| <a href="#">fplot_core::term_get_string_result</a> | 89 |

## 3 Data Type Index

### 3.1 Data Types List

Here are the data types with brief descriptions:

|  |    |
|--|----|
| <a href="#">fplot_core::cm_get_string_result</a><br>Retrieves a string from a colormap                         | 66 |
| <a href="#">fplot_core::color</a><br>Describes an RGB color  | 67 |
| <a href="#">fplot_core::colormap</a><br>A colormap object for a surface plot                                   | 67 |
| <a href="#">fplot_core::cool_colormap</a><br>Defines a colormap consisting of "cool" colors                    | 67 |
| <a href="#">fplot_core::get_string_result</a><br>Retrieves a string from a <a href="#">plot_object</a>         | 68 |
| <a href="#">fplot_core::hot_colormap</a><br>Defines a colormap consisting of "hot" colors                      | 68 |
| <a href="#">fplot_core::legend</a><br>Defines a legend object  | 69 |
| <a href="#">fplot_core::pa_get_string_result</a><br>Retrieves a string from a <a href="#">plot_axis</a>        | 70 |
| <a href="#">fplot_core::pd_get_string_result</a><br>Retrieves a string from a <a href="#">plot_data</a> object | 71 |
| <a href="#">fplot_core::plot</a><br>Defines the basic GNUPLOT plot   | 71 |

|  |    |
|--|----|
| <a href="#">fplot_core::plot_2d</a>  |    |
| A plot object defining a 2D plot   | 73 |
| <a href="#">fplot_core::plot_3d</a>  |    |
| A plot object defining a 3D plot   | 74 |
| <a href="#">fplot_core::plot_axis</a>                                      |    |
| Describes a single plot axis   | 75 |
| <a href="#">fplot_core::plot_data</a>                                      |    |
| Provides a container for plot data   | 77 |
| <a href="#">fplot_core::plot_data_2d</a>                                   |    |
| Defines a two-dimensional plot data set                                    | 77 |
| <a href="#">fplot_core::plot_data_3d</a>                                   |    |
| Defines a three-dimensional plot data set                                  | 79 |
| <a href="#">fplot_core::plot_object</a>                                    |    |
| The base type for a GNUPLOT object   | 80 |
| <a href="#">fplot_core::png_terminal</a>                                   |    |
| Defines a GNUPLOT PNG terminal object                                      | 80 |
| <a href="#">fplot_core::qt_terminal</a>                                    |    |
| Defines a GNUPLOT QT terminal object                                       | 81 |
| <a href="#">fplot_core::rainbow_colormap</a>                               |    |
| Defines a rainbow colormap   | 81 |
| <a href="#">fplot_core::scatter_plot_data</a>                              |    |
| A <a href="#">plot_data</a> object for describing scatter plot data sets   | 82 |
| <a href="#">fplot_core::spd_get_int_value</a>                              |    |
| Retrieves an integer value from a <a href="#">scatter_plot_data</a> object | 84 |
| <a href="#">fplot_core::spd_get_string_result</a>                          |    |
| Retrieves a string from a <a href="#">scatter_plot_data</a> object         | 84 |
| <a href="#">fplot_core::spd_get_value</a>                                  |    |
| Retrieves a numeric value from a <a href="#">scatter_plot_data</a> object  | 85 |
| <a href="#">fplot_core::spd_set_value</a>                                  |    |
| Sets a numeric value into a <a href="#">scatter_plot_data</a> object       | 85 |
| <a href="#">fplot_core::surface_plot</a>                                   |    |
| A plot object defining a 3D surface plot                                   | 86 |
| <a href="#">fplot_core::surface_plot_data</a>                              |    |
| Provides a three-dimensional surface plot data set                         | 87 |
| <a href="#">fplot_core::term_get_string_result</a>                         |    |
| Retrieves a string from a terminal   | 89 |
| <a href="#">fplot_core::terminal</a>                                       |    |
| Defines a GNUPLOT terminal object  | 89 |
| <a href="#">fplot_core::windows_terminal</a>                               |    |
| Defines a GNUPLOT Win32 terminal object                                    | 90 |
| <a href="#">fplot_core::wxt_terminal</a>                                   |    |
| Defines a GNUPLOT WXT terminal object                                      | 91 |

|                                     |    |
|-------------------------------------|----|
| <a href="#">fplot_core::x_axis</a>  |    |
| An x-axis object                    | 92 |
| <a href="#">fplot_core::y2_axis</a> |    |
| A secondary y-axis object           | 92 |
| <a href="#">fplot_core::y_axis</a>  |    |
| A y-axis object                     | 93 |
| <a href="#">fplot_core::z_axis</a>  |    |
| A z-axis object                     | 93 |

## 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](#)  
*A plot object defining a 3D 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 GNUPLLOT 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

|                 |                   |  |
|-----------------|-------------------|--|
| <code>in</code> | <code>this</code> | The <code>cool_colormap</code> object. |
|-----------------|-------------------|--|

#### Returns

The command string.

Definition at line 4441 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

|                      |                   |                    |
|----------------------|-------------------|--------------------|
| <code>in, out</code> | <code>this</code> | The color object.  |
| <code>in</code>      | <code>clr</code>  | 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

|                 |                   |                   |
|-----------------|-------------------|-------------------|
| <code>in</code> | <code>this</code> | 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

|                 |                   |                      |
|-----------------|-------------------|----------------------|
| <code>in</code> | <code>this</code> | The colormap object. |
|-----------------|-------------------|----------------------|

##### Returns

The command string.

Definition at line 4387 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

|                 |                   |  |
|-----------------|-------------------|--|
| <code>in</code> | <code>this</code> | The <a href="#">hot_colormap</a> object. |
|-----------------|-------------------|--|

**Returns**

The command string.

Definition at line 4428 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 | <i>this</i> | 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 | <i>this</i> | 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 | <i>this</i> | 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 | <i>this</i> | The legend object. |
| in      | <i>x</i>    | 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 | <i>this</i> | The legend object.  |
|         | <i>x</i>    | 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 | <i>this</i> | The legend object. |
| in      | <i>x</i>    | 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 | <i>this</i> | The legend object.  |
|         | <i>x</i>    | 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**

|                      |             |                    |
|----------------------|-------------|--------------------|
| <code>in, out</code> | <i>this</i> | The legend object. |
| <code>in</code>      | <i>x</i>    | 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**

|                      |             |                                  |
|----------------------|-------------|----------------------------------|
| <code>in, out</code> | <i>this</i> | The <code>plot_2d</code> object. |
|----------------------|-------------|----------------------------------|

Definition at line 2717 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**

|                 |             |                                  |
|-----------------|-------------|----------------------------------|
| <code>in</code> | <i>this</i> | The <code>plot_2d</code> object. |
|-----------------|-------------|----------------------------------|

**Returns**

The command string.

Definition at line 2798 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**

|                 |             |                                  |
|-----------------|-------------|----------------------------------|
| <code>in</code> | <i>this</i> | The <code>plot_2d</code> object. |
|-----------------|-------------|----------------------------------|

**Returns**

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

Definition at line 2968 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 <a href="#">plot_2d</a> object. |
|----|------|-------------------------------------|

## Returns

A pointer to the x-axis object.

Definition at line 2934 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 <a href="#">plot_2d</a> object. |
|----|------|-------------------------------------|

## Returns

A pointer to the secondary y-axis object.

Definition at line 2956 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 <a href="#">plot_2d</a> object. |
|----|------|-------------------------------------|

## Returns

A pointer to the y-axis object.

Definition at line 2945 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 <a href="#">plot_2d</a> object. |
|----|------|-------------------------------------|



## Parameters

|     |      |  |
|-----|------|--|
| in  | term | An optional input that is used to define the terminal. The default terminal is a WXT terminal. The acceptable inputs are: <ul style="list-style-type: none"> <li>• GNUPLOT_TERMINAL_PNG</li> <li>• GNUPLOT_TERMINAL_QT</li> <li>• GNUPLOT_TERMINAL_WIN32</li> <li>• GNUPLOT_TERMINAL_WXT</li> </ul>  |
| 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. <ul style="list-style-type: none"> <li>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available.</li> </ul> |

Definition at line 2750 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 | <i>this</i> | The <a href="#">plot_2d</a> object.                       |
| in      | <i>x</i>    | Set to true if the axis should be displayed; else, false. |

Definition at line 2980 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 | <i>this</i> | The <a href="#">plot_3d</a> object. |
|---------|-------------|-------------------------------------|

Definition at line 3525 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 | <i>this</i> | The <a href="#">plot_3d</a> object. |
|----|-------------|-------------------------------------|

**Returns**

The azimuth angle, in degrees.

Definition at line 3793 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 | <i>this</i> | The <a href="#">plot_3d</a> object. |
|----|-------------|-------------------------------------|

**Returns**

The command string.

Definition at line 3606 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 | <i>this</i> | The <a href="#">plot_3d</a> object. |
|----|-------------|-------------------------------------|

**Returns**

The elevation angle, in degrees.

Definition at line 3771 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 | <i>this</i> | The <a href="#">plot_3d</a> object. |
|----|-------------|-------------------------------------|

**Returns**

A pointer to the x-axis object.

Definition at line 3738 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**

|                 |                   |                                     |
|-----------------|-------------------|-------------------------------------|
| <code>in</code> | <code>this</code> | The <a href="#">plot_3d</a> object. |
|-----------------|-------------------|-------------------------------------|

**Returns**

A pointer to the y-axis object.

Definition at line 3749 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**

|                 |                   |                                     |
|-----------------|-------------------|-------------------------------------|
| <code>in</code> | <code>this</code> | The <a href="#">plot_3d</a> object. |
|-----------------|-------------------|-------------------------------------|

**Returns**

A pointer to the z-axis object.

Definition at line 3760 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**

|                 |                   |                                     |
|-----------------|-------------------|-------------------------------------|
| <code>in</code> | <code>this</code> | The <a href="#">plot_3d</a> 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 3817 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**

|                 |                   |                                     |
|-----------------|-------------------|-------------------------------------|
| <code>in</code> | <code>this</code> | The <a href="#">plot_3d</a> object. |
|-----------------|-------------------|-------------------------------------|

## Parameters

|     |             |  |
|-----|-------------|--|
| in  | <i>term</i> | An optional input that is used to define the terminal. The default terminal is a WXT terminal. The acceptable inputs are: <ul style="list-style-type: none"> <li>• GNUPLOT_TERMINAL_PNG</li> <li>• GNUPLOT_TERMINAL_QT</li> <li>• GNUPLOT_TERMINAL_WIN32</li> <li>• GNUPLOT_TERMINAL_WXT</li> </ul>  |
| out | <i>err</i>  | 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. <ul style="list-style-type: none"> <li>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available.</li> </ul> |

Definition at line 3558 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 | <i>this</i> | The <a href="#">plot_3d</a> object. |
| in      | <i>x</i>    | The azimuth angle, in degrees.      |

Definition at line 3804 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 | <i>this</i> | The <a href="#">plot_3d</a> object. |
| in      | <i>x</i>    | The elevation angle, in degrees.    |

Definition at line 3782 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**

|                      |                   |   |
|----------------------|-------------------|---|
| <code>in, out</code> | <code>this</code> | The <a href="#">plot_3d</a> object.   |
| <code>in</code>      | <code>x</code>    | 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 3830 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**

|                 |                   |                                       |
|-----------------|-------------------|---------------------------------------|
| <code>in</code> | <code>this</code> | The <a href="#">plot_axis</a> 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**

|                 |                   |                                       |
|-----------------|-------------------|---------------------------------------|
| <code>in</code> | <code>this</code> | The <a href="#">plot_axis</a> 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**

|                 |                   |                                       |
|-----------------|-------------------|---------------------------------------|
| <code>in</code> | <code>this</code> | The <a href="#">plot_axis</a> 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

|                |             |                                       |
|----------------|-------------|---------------------------------------|
| <i>in, out</i> | <i>this</i> | The <a href="#">plot_axis</a> 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

|           |             |                                       |
|-----------|-------------|---------------------------------------|
| <i>in</i> | <i>this</i> | The <a href="#">plot_axis</a> 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

|           |             |                                       |
|-----------|-------------|---------------------------------------|
| <i>in</i> | <i>this</i> | The <a href="#">plot_axis</a> 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 | <i>this</i> | The <a href="#">plot_axis</a> 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 | <i>this</i> | The <a href="#">plot_axis</a> 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 | <i>this</i> | The <a href="#">plot_axis</a> object.                                |
| in      | <i>x</i>    | 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 | <i>this</i>  | The <a href="#">plot_axis</a> object. |
| in      | <i>lower</i> | The lower display limit.              |
| in      | <i>upper</i> | 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 | <i>this</i> | The <a href="#">plot_axis</a> object.                           |
| in      | <i>x</i>    | 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 | <i>this</i> | The <a href="#">plot_axis</a> object.  |
| in      | <i>txt</i>  | 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 | <i>this</i> | The <a href="#">plot_axis</a> object.                   |
| in      | <i>x</i>    | 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 | <i>this</i> | The <a href="#">plot_axis</a> object. |
| in      | <i>x</i>    | 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

|                 |                   |  |
|-----------------|-------------------|--|
| <code>in</code> | <code>this</code> | The <a href="#">plot_data_2d</a> object. |
|-----------------|-------------------|--|

#### Returns

The command string.

Definition at line 3046 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

|                 |                   |  |
|-----------------|-------------------|--|
| <code>in</code> | <code>this</code> | The <a href="#">plot_data_2d</a> object. |
|-----------------|-------------------|--|

#### Returns

The command string.

Definition at line 3065 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

|                 |                   |  |
|-----------------|-------------------|--|
| <code>in</code> | <code>this</code> | The <a href="#">plot_data_2d</a> object. |
|-----------------|-------------------|--|

#### Returns

The number of data points.

Definition at line 3098 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 | <i>this</i> | The <a href="#">plot_data_2d</a> 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 3234 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 | <i>this</i>  | The <a href="#">plot_data_2d</a> object. |
| in | <i>index</i> | The index of the data point to retrieve. |

## Returns

The requested data point.

Definition at line 3114 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 | <i>this</i>  | The <a href="#">plot_data_2d</a> object. |
| in | <i>index</i> | The index of the data point to retrieve. |

## Returns

The requested data point.

Definition at line 3146 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 | <i>this</i> | The <a href="#">plot_data_2d</a> object.  |
| in      | <i>x</i>    | An N-element array containing the x coordinate data.  |
| in      | <i>y</i>    | An N-element array containing the y coordinate data.  |
| out     | <i>err</i>  | <p>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.</p> <ul style="list-style-type: none"> <li>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available.</li> <li>• PLOT_ARRAY_SIZE_MISMATCH_ERROR: Occurs if <i>x</i> and <i>y</i> are not the same size.</li> </ul> |

Definition at line 3186 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 | <i>this</i> | The <a href="#">plot_data_2d</a> object.  |
| in      | <i>y</i>    | An N-element array containing the y-coordinate data. This data will be plotted against its own index.   |
| out     | <i>err</i>  | <p>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.</p> <ul style="list-style-type: none"> <li>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available.</li> </ul> |

Definition at line 3265 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 | <i>this</i> | The <a href="#">plot_data_2d</a> object.  |
| in      | <i>x</i>    | 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 3247 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 | <i>this</i>  | The <a href="#">plot_data_2d</a> object. |
| in      | <i>index</i> | The index of the data point to replace.  |
| in      | <i>x</i>     | The data point.                          |

Definition at line 3131 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 | <i>this</i>  | The <a href="#">plot_data_2d</a> object. |
| in      | <i>index</i> | The index of the data point to replace.  |
| in      | <i>x</i>     | The data point.                          |

Definition at line 3163 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 | <i>this</i> | The <a href="#">plot_data_3d</a> object. |
|----|-------------|--|

## Returns

The command string.

Definition at line 3417 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 | <i>this</i> | The <a href="#">plot_data_3d</a> object. |
|----|-------------|--|

## Returns

The command string.

Definition at line 3432 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 | <i>this</i> | The <a href="#">plot_data_3d</a> object. |
|----|-------------|--|

#### Returns

The number of data points.

Definition at line 3305 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 | <i>this</i>  | The <a href="#">plot_data_3d</a> object. |
| in | <i>index</i> | The index of the data point to retrieve. |

#### Returns

The requested data point.

Definition at line 3321 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 | <i>this</i>  | The <a href="#">plot_data_3d</a> object. |
| in | <i>index</i> | The index of the data point to retrieve. |

#### Returns

The requested data point.

Definition at line 3353 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 | <i>this</i>  | The <a href="#">plot_data_3d</a> object. |
| in | <i>index</i> | The index of the data point to retrieve. |

## Returns

The requested data point.

Definition at line 3385 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 | <i>this</i> | The <a href="#">plot_data_2d</a> object.  |
| in      | <i>x</i>    | An N-element array containing the x coordinate data.  |
| in      | <i>y</i>    | An N-element array containing the y coordinate data.  |
| in      | <i>z</i>    | An N-element array containing the z coordinate data.  |
| out     | <i>err</i>  | 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. <ul style="list-style-type: none"> <li>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available.</li> <li>• PLOT_ARRAY_SIZE_MISMATCH_ERROR: Occurs if <i>x</i>, <i>y</i>, and <i>z</i> are not the same size.</li> </ul> |

Definition at line 3477 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 | <i>this</i>  | The <a href="#">plot_data_3d</a> object. |
| in      | <i>index</i> | The index of the data point to replace.  |
| in      | <i>x</i>     | The data point.                          |

Definition at line 3338 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 | <i>this</i>  | The <a href="#">plot_data_3d</a> object. |
| in      | <i>index</i> | The index of the data point to replace.  |
| in      | <i>x</i>     | The data point.                          |

Definition at line 3370 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 | <i>this</i>  | The <a href="#">plot_data_3d</a> object. |
| in      | <i>index</i> | The index of the data point to replace.  |
| in      | <i>x</i>     | The data point.                          |

Definition at line 3402 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 | <i>this</i> | The <a href="#">plot_data</a> 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 | <i>this</i> | The <a href="#">plot_data</a> object. |
| in      | <i>txt</i>  | 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 | <i>this</i> | 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 | <i>this</i> | 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  | <i>this</i>    | The plot object.  |
| in  | <i>persist</i> | 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.  |
| out | <i>err</i>     | 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. <ul style="list-style-type: none"> <li>• PLOT_GNUPLOT_FILE_ERROR: Occurs if the command file cannot be written.</li> </ul> |

Definition at line 2173 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 | <i>this</i> | The plot object.                                   |
| in | <i>i</i>    | The index of the <a href="#">plot_data</a> 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**

|                 |                   |                  |
|-----------------|-------------------|------------------|
| <code>in</code> | <code>this</code> | 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**

|                 |                   |                  |
|-----------------|-------------------|------------------|
| <code>in</code> | <code>this</code> | The plot object. |
|-----------------|-------------------|------------------|

**Returns**

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

Definition at line 2355 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**

|                 |                   |                  |
|-----------------|-------------------|------------------|
| <code>in</code> | <code>this</code> | The plot object. |
|-----------------|-------------------|------------------|

**Returns**

The font name.

Definition at line 2275 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 | <i>this</i> | The plot object. |
|----|-------------|------------------|

**Returns**

The size of the font, in points.

Definition at line 2301 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 | <i>this</i> | 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 | <i>this</i> | The plot object. |
|----|-------------|------------------|

**Returns**

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

Definition at line 2142 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 | <i>this</i> | The plot object. |
|----|-------------|------------------|

**Returns**

A pointer to the GNUPLOT terminal object.

Definition at line 2131 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 | <i>this</i> | 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 2331 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 | <i>this</i> | 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 | <i>this</i> | 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

|                |             |   |
|----------------|-------------|---|
| <i>in, out</i> | <i>this</i> | The plot object.  |
| <i>in</i>      | <i>term</i> | <p>An optional input that is used to define the terminal. The default terminal is a WXT terminal. The acceptable inputs are:</p> <ul style="list-style-type: none"> <li>• GNUPLOT_TERMINAL_PNG</li> <li>• GNUPLOT_TERMINAL_QT</li> <li>• GNUPLOT_TERMINAL_WIN32</li> <li>• GNUPLOT_TERMINAL_WXT</li> </ul>  |
| <i>out</i>     | <i>err</i>  | <p>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.</p> <ul style="list-style-type: none"> <li>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available.</li> </ul> |

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

|                |             |                  |
|----------------|-------------|------------------|
| <i>in, out</i> | <i>this</i> | 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

|                |             |   |
|----------------|-------------|---|
| <i>in, out</i> | <i>this</i> | The plot object.  |
| <i>in</i>      | <i>x</i>    | The <code>plot_data</code> object.  |
| <i>out</i>     | <i>err</i>  | <p>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.</p> <ul style="list-style-type: none"> <li>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available.</li> </ul> |

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  | <i>this</i>  | The plot object.  |
| in  | <i>fname</i> | The filename.   |
| out | <i>err</i>   | 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. <ul style="list-style-type: none"> <li>PLOT_GNUPLOT_FILE_ERROR: Occurs if the command file cannot be written.</li> </ul> |

Definition at line 2237 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 | <i>this</i> | The plot object.                                |
| in      | <i>i</i>    | The index of the <code>plot_data</code> object. |
| in      | <i>x</i>    | The <code>plot_data</code> object.              |

Definition at line 2119 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 | <i>this</i> | The plot object.  |
| in      | <i>x</i>    | Set to true if the border should be drawn; else, false. |

Definition at line 2366 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 | <i>this</i> | The plot object. |
| in      | <i>x</i>    | The font name.   |

Definition at line 2288 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 | <i>this</i> | The plot object.  |
| in      | <i>x</i>    | 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 2316 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 | <i>this</i> | The plot object.  |
| in      | <i>x</i>    | Set to true if the grid lines should be shown; else, false. |

Definition at line 2153 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 | <i>this</i> | The plot object.   |
| in      | <i>x</i>    | Set to true if the tic marks should point inwards; else, false if the tic marks should point outwards. |

Definition at line 2344 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 | <i>this</i> | The plot object.   |
| in      | <i>txt</i>  | 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

|                 |                   |                      |
|-----------------|-------------------|----------------------|
| <code>in</code> | <code>this</code> | 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

|                 |                   |  |
|-----------------|-------------------|--|
| <code>in</code> | <code>this</code> | The <a href="#">png_terminal</a> 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

|                 |                   |  |
|-----------------|-------------------|--|
| <code>in</code> | <code>this</code> | The <a href="#">png_terminal</a> 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 | <i>this</i> | The <a href="#">png_terminal</a> object.       |
| in      | <i>The</i>  | 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 | <i>this</i> | The <a href="#">qt_terminal</a> 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 | <i>this</i> | The <a href="#">rainbow_colormap</a> object. |
|----|-------------|--|

**Returns**

The command string.

Definition at line 4414 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object. |
|----|-------------|---|

**Returns**

The command string.

Definition at line 2380 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object. |
|----|-------------|---|

**Returns**

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

Definition at line 2539 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object. |
|----|-------------|---|

**Returns**

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

Definition at line 2561 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object. |
|----|-------------|---|

**Returns**

The color.

Definition at line 2517 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> 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 2482 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object. |
|----|-------------|---|

#### Returns

The line width.

Definition at line 2455 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object. |
|----|-------------|---|

#### Returns

The marker frequency.

Definition at line 2670 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 <a href="#">scatter_plot_data</a> object. |
|----|------|---|

#### Returns

The scaling factor.

Definition at line 2648 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 <a href="#">scatter_plot_data</a> 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 2596 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object. |
|----|-------------|---|

**Returns**

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

Definition at line 2693 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object.         |
| in      | <i>x</i>    | Set to true if the line should be drawn; else, false. |

Definition at line 2550 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object.            |
| in      | <i>x</i>    | Set to true if the markers should be drawn; else, false. |

Definition at line 2572 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object. |
| in      | <i>x</i>    | The color.                                    |

Definition at line 2528 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object.  |
| in      | <i>x</i>    | The line style. The line style must be one of the following: <ul style="list-style-type: none"> <li>• LINE_DASHED</li> <li>• LINE_DASH_DOTTED</li> <li>• LINE_DASH_DOT_DOT</li> <li>• LINE_DOTTED</li> <li>• LINE_SOLID</li> </ul> |

Definition at line 2499 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object. |
| in      | <i>x</i>    | The line width.                               |

Definition at line 2466 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object. |
| in      | <i>x</i>    | The marker frequency.                         |

Definition at line 2681 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object. |
| in      | <i>x</i>    | The scaling factor.                           |

Definition at line 2659 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object.   |
| in      | <i>x</i>    | <p>The marker type. The marker type must be one of the following:</p> <ul style="list-style-type: none"> <li>• MARKER_ASTERISK</li> <li>• MARKER_EMPTY_CIRCLE</li> <li>• MARKER_EMPTY_NABLA</li> <li>• MARKER_EMPTY_RHOMBUS</li> <li>• MARKER_EMPTY_SQUARE</li> <li>• MARKER_EMPTY_TRIANGLE</li> <li>• MARKER_FILLED_CIRCLE</li> <li>• MARKER_FILLED_NABLA</li> <li>• MARKER_FILLED_RHOMBUS</li> <li>• MARKER_FILLED_SQUARE</li> <li>• MARKER_FILLED_TRIANGLE</li> <li>• MARKER_PLUS</li> <li>• MARKER_X</li> </ul> |

Definition at line 2621 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 | <i>this</i> | The <a href="#">scatter_plot_data</a> object.             |
| in      | <i>x</i>    | Set to true if GNUPLOT should choose colors; else, false. |

Definition at line 2705 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 <a href="#">surface_plot</a> object. |
|---------|------|--|

Definition at line 4131 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 <a href="#">surface_plot</a> object. |
|----|------|--|

## Returns

The command string.

Definition at line 4203 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 <a href="#">surface_plot</a> object. |
|----|------|--|

## Returns

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

Definition at line 4261 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 <a href="#">surface_plot</a> object. |
|----|------|--|

## Returns

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

Definition at line 4363 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 <a href="#">surface_plot</a> object. |
|----|------|--|

#### Returns

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

Definition at line 4339 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 <a href="#">surface_plot</a> object. |
|----|------|--|

#### Returns

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

Definition at line 4180 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 <a href="#">surface_plot</a> object. |
|----|------|--|

#### Returns

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

Definition at line 4314 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  | <i>this</i> | The <a href="#">surface_plot</a> object.   |
| in  | <i>term</i> | An optional input that is used to define the terminal. The default terminal is a WXT terminal. The acceptable inputs are: <ul style="list-style-type: none"> <li>• GNUPLOT_TERMINAL_PNG</li> <li>• GNUPLOT_TERMINAL_QT</li> <li>• GNUPLOT_TERMINAL_WIN32</li> <li>• GNUPLOT_TERMINAL_WXT</li> </ul>  |
| out | <i>err</i>  | 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. <ul style="list-style-type: none"> <li>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available.</li> </ul> |

Definition at line 4158 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 | <i>this</i> | The <a href="#">surface_plot</a> object.   |
| in      | <i>x</i>    | The colormap object. Notice, a copy of this object is stored, and the <a href="#">surface_plot</a> object then manages the lifetime of the copy.   |
| out     | <i>err</i>  | 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. <ul style="list-style-type: none"> <li>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available.</li> </ul> |

Definition at line 4280 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 | <i>this</i> | The <a href="#">surface_plot</a> object.                  |
| in      | <i>x</i>    | Set to true if the colorbar should be drawn; else, false. |

Definition at line 4374 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 | <i>this</i> | The <a href="#">surface_plot</a> object.   |
| in      | <i>x</i>    | Set to true if the contour plot should be drawn; else, false to only draw the surface. |

Definition at line 4352 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 | <i>this</i> | The <a href="#">surface_plot</a> object.                  |
| in      | <i>x</i>    | Set to true if hidden lines should be shown; else, false. |

Definition at line 4191 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 | <i>this</i> | The <a href="#">surface_plot</a> object.                    |
| in      | <i>x</i>    | Set to true if the surface should be smoothed; else, false. |

Definition at line 4326 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 | <i>this</i> | The <a href="#">surface_plot_data</a> object. |
|----|-------------|---|

#### Returns

The command string.

Definition at line 3988 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 | <i>this</i> | The <a href="#">surface_plot_data</a> object. |
|----|-------------|---|

#### Returns

The GNUPLOT command string.

Definition at line 4027 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 | <i>this</i> | The <a href="#">surface_plot_data</a> object.  |
| in | <i>dim</i>  | 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 3845 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 | <i>this</i> | The <a href="#">surface_plot_data</a> object. |
|----|-------------|---|

#### Returns

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

Definition at line 3964 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 | <i>this</i> | The <a href="#">surface_plot_data</a> object. |
| in | <i>i</i>    | The row index.                                |
| in | <i>j</i>    | The column index.                             |

**Returns**

The value.

Definition at line 3863 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 | <i>this</i> | The <a href="#">surface_plot_data</a> object. |
| in | <i>i</i>    | The row index.                                |
| in | <i>j</i>    | The column index.                             |

**Returns**

The value.

Definition at line 3897 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 | <i>this</i> | The <a href="#">surface_plot_data</a> object. |
| in | <i>i</i>    | The row index.                                |
| in | <i>j</i>    | The column index.                             |

**Returns**

The value.

Definition at line 3931 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 | <i>this</i> | The <a href="#">plot_data_2d</a> object.   |
| in      | <i>x</i>    | An M-by-N matrix containing the x-coordinate data.   |
| in      | <i>y</i>    | An M-by-N matrix containing the y-coordinate data.   |
| in      | <i>z</i>    | An M-by-N matrix containing the z-coordinate data.   |
| out     | <i>err</i>  | <p>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.</p> <ul style="list-style-type: none"> <li>• PLOT_OUT_OF_MEMORY_ERROR: Occurs if insufficient memory is available.</li> <li>• PLOT_ARRAY_SIZE_MISMATCH_ERROR: Occurs if <i>x</i>, <i>y</i>, and <i>z</i> are not the same size.</li> </ul> |

Definition at line 4076 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 | <i>this</i> | The <a href="#">surface_plot_data</a> object.  |
| in      | <i>x</i>    | Set to true if a wireframe mesh should be displayed; else, false to display a solid surface. |

Definition at line 3976 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 | <i>this</i> | The <a href="#">surface_plot_data</a> object. |
| in      | <i>i</i>    | The row index.                                |
| in      | <i>j</i>    | The column index.                             |
| in      | <i>x</i>    | The value.                                    |

Definition at line 3881 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 | <i>this</i> | The <a href="#">surface_plot_data</a> object. |
| in      | <i>i</i>    | The row index.                                |
| in      | <i>j</i>    | The column index.                             |
| in      | <i>x</i>    | The value.                                    |

Definition at line 3915 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 | <i>this</i> | The <a href="#">surface_plot_data</a> object. |
| in      | <i>i</i>    | The row index.                                |
| in      | <i>j</i>    | The column index.                             |
| in      | <i>x</i>    | The value.                                    |

Definition at line 3949 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 | <i>this</i> | 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 | <i>this</i> | 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**

|                 |                          |                      |
|-----------------|--------------------------|----------------------|
| <code>in</code> | <code><i>this</i></code> | 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**

|                 |                          |                      |
|-----------------|--------------------------|----------------------|
| <code>in</code> | <code><i>this</i></code> | 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**

|                 |                          |                      |
|-----------------|--------------------------|----------------------|
| <code>in</code> | <code><i>this</i></code> | 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 | <i>this</i> | 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 | <i>this</i> | 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 | <i>this</i> | The terminal object.   |
| in      | <i>name</i> | 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 | <i>this</i> | The terminal object.  |
| in      | <i>sz</i>   | 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 | <i>this</i> | The terminal object.    |
| in      | <i>x</i>    | 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 | <i>this</i> | The terminal object. |
| in      | <i>txt</i>  | 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 | <i>this</i> | The terminal object.   |
| in      | <i>x</i>    | 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 | <i>this</i> | The terminal object.   |
| in      | <i>x</i>    | 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 | <i>this</i> | The <a href="#">windows_terminal</a> 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 | <i>this</i> | The <a href="#">wxt_terminal</a> 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 | <i>this</i> | The <a href="#">x_axis</a> object. |
|----|-------------|------------------------------------|

#### Returns

The string.

Definition at line 2993 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 <a href="#">y2_axis</a> object. |
|----|------|-------------------------------------|

## Returns

The string.

Definition at line 3019 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 <a href="#">y_axis</a> object. |
|----|------|------------------------------------|

## Returns

The string.

Definition at line 3006 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 <a href="#">z_axis</a> object. |
|----|------|------------------------------------|

## Returns

The string.

Definition at line 3032 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.

## 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

|                 |                   |                      |
|-----------------|-------------------|----------------------|
| <code>in</code> | <code>this</code> | 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\_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.*

### 5.4.1 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`

## 5.5 `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)

### 5.5.1 Detailed Description

Retrieves a string from a `plot_object`.

#### Parameters

|                 |                   |                                      |
|-----------------|-------------------|--------------------------------------|
| <code>in</code> | <code>this</code> | The <code>plot_object</code> 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`

## 5.6 `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.*

### 5.6.1 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

## 5.7 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.*

### 5.7.1 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`

## 5.8 `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)

### 5.8.1 Detailed Description

Retrieves a string from a `plot_axis`.

#### Parameters

|                 |                   |                                    |
|-----------------|-------------------|------------------------------------|
| <code>in</code> | <code>this</code> | The <code>plot_axis</code> 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`

## 5.9 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)

### 5.9.1 Detailed Description

Retrieves a string from a [plot\\_data](#) object.

#### Parameters

|                 |                   |                                       |
|-----------------|-------------------|---------------------------------------|
| <code>in</code> | <code>this</code> | The <a href="#">plot_data</a> 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`

## 5.10 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?*

### 5.10.1 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`

## 5.11 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?*

#### 5.11.1 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`

## 5.12 `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?*

## 5.12.1 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`

## 5.13 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.*

### 5.13.1 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

## 5.14 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.*

### 5.14.1 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

## 5.15 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?*

### 5.15.1 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`

## 5.16 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.*

#### 5.16.1 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`

## 5.17 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.*

### 5.17.1 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`

## 5.18 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.*

### 5.18.1 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

## 5.19 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.*

### 5.19.1 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

## 5.20 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.*

### 5.20.1 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`

### 5.21 `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.*

### 5.21.1 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`

## 5.22 `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)

### 5.22.1 Detailed Description

Retrieves an integer value from a [scatter\\_plot\\_data](#) object.

#### Parameters

|    |             |   |
|----|-------------|---|
| in | <i>this</i> | The <a href="#">scatter_plot_data</a> 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`

## 5.23 `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)

### 5.23.1 Detailed Description

Retrieves a string from a [scatter\\_plot\\_data](#) object.

## Parameters

|    |             |   |
|----|-------------|---|
| in | <i>this</i> | The <a href="#">scatter_plot_data</a> 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`

## 5.24 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*)

## 5.24.1 Detailed Description

Retrieves a numeric value from a [scatter\\_plot\\_data](#) object.

## Parameters

|    |              |   |
|----|--------------|---|
| in | <i>this</i>  | The <a href="#">scatter_plot_data</a> object. |
| in | <i>index</i> | 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`

## 5.25 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)

### 5.25.1 Detailed Description

Sets a numeric value into a [scatter\\_plot\\_data](#) object.

#### Parameters

|         |              |   |
|---------|--------------|---|
| in, out | <i>this</i>  | The <a href="#">scatter_plot_data</a> object. |
| in      | <i>index</i> | The index of the value to retrieve.           |
| in      | <i>x</i>     | 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`

## 5.26 `fplot_core::surface_plot` Type Reference

A plot object defining a 3D surface plot.

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?*

#### 5.26.1 Detailed Description

A plot object defining a 3D surface plot.

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`

## 5.27 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 GNU PLOT command string to represent this [surface\\_plot\\_data](#) object.*
- procedure, public `get_data_string` => `surfd_get_data_cmd`  
*Gets the GNU PLOT 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.*

### 5.27.1 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`

## 5.28 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)

### 5.28.1 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

## 5.29 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.*

#### 5.29.1 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`

#### 5.30 `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.*

##### 5.30.1 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`

## 5.31 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.*

##### 5.31.1 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`

### 5.32 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.*

#### 5.32.1 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

### 5.33 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.*

#### 5.33.1 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

### 5.34 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.*

#### 5.34.1 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

### 5.35 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.*

#### 5.35.1 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, 15

clr\_copy\_from  
    fplot\_core, 15

clr\_to\_hex\_string  
    fplot\_core, 16

cm\_get\_cmd  
    fplot\_core, 16

fplot\_core, 5

    ccm\_get\_clr, 15

    clr\_copy\_from, 15

    clr\_to\_hex\_string, 16

    cm\_get\_cmd, 16

    hcm\_get\_clr, 16

    leg\_get\_box, 17

    leg\_get\_command\_txt, 17

    leg\_get\_horz\_pos, 17

    leg\_get\_inside, 17

    leg\_get\_vert\_pos, 18

    leg\_get\_visible, 18

    leg\_set\_box, 18

    leg\_set\_horz\_pos, 19

    leg\_set\_inside, 19

    leg\_set\_vert\_pos, 19

    leg\_set\_visible, 19

    p2d\_clean\_up, 20

    p2d\_get\_cmd, 20

    p2d\_get\_use\_y2, 20

    p2d\_get\_x\_axis, 20

    p2d\_get\_y2\_axis, 21

    p2d\_get\_y\_axis, 21

    p2d\_init, 21

    p2d\_set\_use\_y2, 22

    p3d\_clean\_up, 22

    p3d\_get\_azimuth, 22

    p3d\_get\_cmd, 23

    p3d\_get\_elevation, 23

    p3d\_get\_x\_axis, 23

    p3d\_get\_y\_axis, 23

    p3d\_get\_z\_axis, 24

    p3d\_get\_z\_axis\_intersect, 24

    p3d\_init, 24

    p3d\_set\_azimuth, 25

    p3d\_set\_elevation, 25

    p3d\_set\_z\_axis\_intersect, 25

    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, 27

    pa\_has\_title, 28

    pa\_set\_autoscale, 28

    pa\_set\_axis\_limits, 28

    pa\_set\_log\_scale, 28

    pa\_set\_title, 29

    pa\_set\_zero\_axis, 29

    pa\_set\_zero\_axis\_width, 29

    pd2d\_get\_axes\_cmd, 29

    pd2d\_get\_data\_cmd, 30

    pd2d\_get\_data\_count, 30

    pd2d\_get\_draw\_against\_y2, 30

    pd2d\_get\_x\_data, 31

    pd2d\_get\_y\_data, 31

    pd2d\_set\_data\_1, 31

    pd2d\_set\_data\_2, 32

    pd2d\_set\_draw\_against\_y2, 32

    pd2d\_set\_x\_data, 32

    pd2d\_set\_y\_data, 33

    pd3d\_get\_axes\_cmd, 33

    pd3d\_get\_data\_cmd, 33

    pd3d\_get\_data\_count, 33

    pd3d\_get\_x\_data, 34

    pd3d\_get\_y\_data, 34

    pd3d\_get\_z\_data, 34

    pd3d\_set\_data\_1, 35

    pd3d\_set\_x\_data, 35

    pd3d\_set\_y\_data, 35

    pd3d\_set\_z\_data, 36

    pd\_get\_name, 36

    pd\_set\_name, 36

    plt\_clean\_up, 37

    plt\_clear\_all, 37

    plt\_draw, 37

    plt\_get, 37

    plt\_get\_count, 38

    plt\_get\_draw\_border, 38

    plt\_get\_font, 38

    plt\_get\_font\_size, 38

    plt\_get\_legend, 39

    plt\_get\_show\_grid, 39

    plt\_get\_term, 39

    plt\_get\_ticks\_in, 40

    plt\_get\_title, 40

    plt\_has\_title, 40

    plt\_init, 40

    plt\_pop\_data, 42

    plt\_push\_data, 42

    plt\_save, 42

    plt\_set, 43

    plt\_set\_draw\_border, 43

    plt\_set\_font, 43

    plt\_set\_font\_size, 44

    plt\_set\_show\_grid, 44

    plt\_set\_ticks\_in, 44

    plt\_set\_title, 44

    png\_get\_command\_string, 44

    png\_get\_filename, 45

png\_get\_term\_string, 45  
 png\_set\_filename, 45  
 qt\_get\_term\_string, 46  
 rcm\_get\_clr, 46  
 spd\_get\_cmd, 46  
 spd\_get\_draw\_line, 47  
 spd\_get\_draw\_markers, 47  
 spd\_get\_line\_color, 47  
 spd\_get\_line\_style, 47  
 spd\_get\_line\_width, 48  
 spd\_get\_marker\_frequency, 48  
 spd\_get\_marker\_scaling, 48  
 spd\_get\_marker\_style, 49  
 spd\_get\_use\_auto\_colors, 49  
 spd\_set\_draw\_line, 50  
 spd\_set\_draw\_markers, 50  
 spd\_set\_line\_color, 50  
 spd\_set\_line\_style, 50  
 spd\_set\_line\_width, 51  
 spd\_set\_marker\_frequency, 51  
 spd\_set\_marker\_scaling, 51  
 spd\_set\_marker\_style, 52  
 spd\_set\_use\_auto\_colors, 52  
 surf\_clean\_up, 52  
 surf\_get\_cmd, 53  
 surf\_get\_colormap, 53  
 surf\_get\_show\_colorbar, 53  
 surf\_get\_show\_contours, 53  
 surf\_get\_show\_hidden, 54  
 surf\_get\_smooth, 54  
 surf\_init, 54  
 surf\_set\_colormap, 55  
 surf\_set\_show\_colorbar, 55  
 surf\_set\_show\_contours, 55  
 surf\_set\_show\_hidden, 56  
 surf\_set\_smooth, 56  
 surfd\_get\_cmd, 56  
 surfd\_get\_data\_cmd, 56  
 surfd\_get\_size, 57  
 surfd\_get\_wireframe, 57  
 surfd\_get\_x, 57  
 surfd\_get\_y, 58  
 surfd\_get\_z, 58  
 surfd\_set\_data\_1, 58  
 surfd\_set\_wireframe, 59  
 surfd\_set\_x, 59  
 surfd\_set\_y, 59  
 surfd\_set\_z, 60  
 term\_get\_command\_string, 60  
 term\_get\_font\_name, 60  
 term\_get\_font\_size, 61  
 term\_get\_plot\_window\_number, 61  
 term\_get\_title, 61  
 term\_get\_window\_height, 61  
 term\_get\_window\_width, 62  
 term\_set\_font\_name, 62  
 term\_set\_font\_size, 62  
 term\_set\_plot\_window\_number, 63  
 term\_set\_title, 63  
 term\_set\_window\_height, 63  
 term\_set\_window\_width, 63  
 wt\_get\_term\_string, 64  
 wxt\_get\_term\_string, 64  
 xa\_get\_id, 64  
 y2a\_get\_id, 64  
 ya\_get\_id, 65  
 za\_get\_id, 65  
 fplot\_core::cm\_get\_string\_result, 66  
 fplot\_core::color, 67  
 fplot\_core::colormap, 67  
 fplot\_core::cool\_colormap, 67  
 fplot\_core::get\_string\_result, 68  
 fplot\_core::hot\_colormap, 68  
 fplot\_core::legend, 69  
 fplot\_core::pa\_get\_string\_result, 70  
 fplot\_core::pd\_get\_string\_result, 71  
 fplot\_core::plot, 71  
 fplot\_core::plot\_2d, 73  
 fplot\_core::plot\_3d, 74  
 fplot\_core::plot\_axis, 75  
 fplot\_core::plot\_data, 77  
 fplot\_core::plot\_data\_2d, 77  
 fplot\_core::plot\_data\_3d, 79  
 fplot\_core::plot\_object, 80  
 fplot\_core::png\_terminal, 80  
 fplot\_core::qt\_terminal, 81  
 fplot\_core::rainbow\_colormap, 81  
 fplot\_core::scatter\_plot\_data, 82  
 fplot\_core::spd\_get\_int\_value, 84  
 fplot\_core::spd\_get\_string\_result, 84  
 fplot\_core::spd\_get\_value, 85  
 fplot\_core::spd\_set\_value, 85  
 fplot\_core::surface\_plot, 86  
 fplot\_core::surface\_plot\_data, 87  
 fplot\_core::term\_get\_string\_result, 89  
 fplot\_core::terminal, 89  
 fplot\_core::windows\_terminal, 90  
 fplot\_core::wxt\_terminal, 91  
 fplot\_core::x\_axis, 92  
 fplot\_core::y2\_axis, 92  
 fplot\_core::y\_axis, 93  
 fplot\_core::z\_axis, 93  
 fplot\_errors, 65  
 hcm\_get\_clr  
     fplot\_core, 16  
 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, 17  
 leg\_get\_vert\_pos  
     fplot\_core, 18

leg\_get\_visible  
    fplot\_core, 18

leg\_set\_box  
    fplot\_core, 18

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, 19

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, 20

p2d\_get\_y2\_axis  
    fplot\_core, 21

p2d\_get\_y\_axis  
    fplot\_core, 21

p2d\_init  
    fplot\_core, 21

p2d\_set\_use\_y2  
    fplot\_core, 22

p3d\_clean\_up  
    fplot\_core, 22

p3d\_get\_azimuth  
    fplot\_core, 22

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, 23

p3d\_get\_z\_axis  
    fplot\_core, 24

p3d\_get\_z\_axis\_intersect  
    fplot\_core, 24

p3d\_init  
    fplot\_core, 24

p3d\_set\_azimuth  
    fplot\_core, 25

p3d\_set\_elevation  
    fplot\_core, 25

p3d\_set\_z\_axis\_intersect  
    fplot\_core, 25

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, 27

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, 28

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, 29

pd2d\_get\_data\_cmd  
    fplot\_core, 30

pd2d\_get\_data\_count  
    fplot\_core, 30

pd2d\_get\_draw\_against\_y2  
    fplot\_core, 30

pd2d\_get\_x\_data  
    fplot\_core, 31

pd2d\_get\_y\_data  
    fplot\_core, 31

pd2d\_set\_data\_1  
    fplot\_core, 31

pd2d\_set\_data\_2  
    fplot\_core, 32

pd2d\_set\_draw\_against\_y2  
    fplot\_core, 32

pd2d\_set\_x\_data  
    fplot\_core, 32

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, 33

pd3d\_get\_x\_data  
    fplot\_core, 34

pd3d\_get\_y\_data  
    fplot\_core, 34

pd3d\_get\_z\_data  
    fplot\_core, 34

pd3d\_set\_data\_1  
    fplot\_core, 35

pd3d\_set\_x\_data  
    fplot\_core, 35

pd3d\_set\_y\_data  
    fplot\_core, 35

pd3d\_set\_z\_data  
    fplot\_core, 36

pd\_get\_name  
    fplot\_core, 36

pd\_set\_name  
    fplot\_core, 36

plt\_clean\_up  
    fplot\_core, 37

plt\_clear\_all  
    fplot\_core, 37

plt\_draw  
    fplot\_core, 37

plt\_get  
    fplot\_core, 37

plt\_get\_count  
    fplot\_core, 38

plt\_get\_draw\_border  
    fplot\_core, 38

plt\_get\_font  
    fplot\_core, 38

plt\_get\_font\_size  
    fplot\_core, 38

plt\_get\_legend  
    fplot\_core, 39

plt\_get\_show\_grid  
    fplot\_core, 39

plt\_get\_term  
    fplot\_core, 39

plt\_get\_ticks\_in  
    fplot\_core, 40

plt\_get\_title  
    fplot\_core, 40

plt\_has\_title  
    fplot\_core, 40

plt\_init  
    fplot\_core, 40

plt\_pop\_data  
    fplot\_core, 42

plt\_push\_data  
    fplot\_core, 42

plt\_save  
    fplot\_core, 42

plt\_set  
    fplot\_core, 43

plt\_set\_draw\_border  
    fplot\_core, 43

plt\_set\_font  
    fplot\_core, 43

plt\_set\_font\_size  
    fplot\_core, 44

plt\_set\_show\_grid  
    fplot\_core, 44

plt\_set\_ticks\_in  
    fplot\_core, 44

plt\_set\_title  
    fplot\_core, 44

png\_get\_command\_string  
    fplot\_core, 44

png\_get\_filename  
    fplot\_core, 45

png\_get\_term\_string  
    fplot\_core, 45

png\_set\_filename  
    fplot\_core, 45

qt\_get\_term\_string  
    fplot\_core, 46

rcm\_get\_clr  
    fplot\_core, 46

spd\_get\_cmd  
    fplot\_core, 46

spd\_get\_draw\_line  
    fplot\_core, 47

spd\_get\_draw\_markers  
    fplot\_core, 47

spd\_get\_line\_color  
    fplot\_core, 47

spd\_get\_line\_style  
    fplot\_core, 47

spd\_get\_line\_width  
    fplot\_core, 48

spd\_get\_marker\_frequency  
    fplot\_core, 48

spd\_get\_marker\_scaling  
    fplot\_core, 48

spd\_get\_marker\_style  
    fplot\_core, 49

spd\_get\_use\_auto\_colors  
    fplot\_core, 49

spd\_set\_draw\_line  
    fplot\_core, 50

spd\_set\_draw\_markers  
    fplot\_core, 50

spd\_set\_line\_color  
    fplot\_core, 50

spd\_set\_line\_style  
    fplot\_core, 50

spd\_set\_line\_width  
    fplot\_core, 51

spd\_set\_marker\_frequency  
    fplot\_core, 51

spd\_set\_marker\_scaling  
    fplot\_core, 51

spd\_set\_marker\_style  
    fplot\_core, 52

spd\_set\_use\_auto\_colors  
    fplot\_core, 52

surf\_clean\_up  
    fplot\_core, 52

surf\_get\_cmd  
    fplot\_core, 53

surf\_get\_colormap  
    fplot\_core, [53](#)  
surf\_get\_show\_colorbar  
    fplot\_core, [53](#)  
surf\_get\_show\_contours  
    fplot\_core, [53](#)  
surf\_get\_show\_hidden  
    fplot\_core, [54](#)  
surf\_get\_smooth  
    fplot\_core, [54](#)  
surf\_init  
    fplot\_core, [54](#)  
surf\_set\_colormap  
    fplot\_core, [55](#)  
surf\_set\_show\_colorbar  
    fplot\_core, [55](#)  
surf\_set\_show\_contours  
    fplot\_core, [55](#)  
surf\_set\_show\_hidden  
    fplot\_core, [56](#)  
surf\_set\_smooth  
    fplot\_core, [56](#)  
surfd\_get\_cmd  
    fplot\_core, [56](#)  
surfd\_get\_data\_cmd  
    fplot\_core, [56](#)  
surfd\_get\_size  
    fplot\_core, [57](#)  
surfd\_get\_wireframe  
    fplot\_core, [57](#)  
surfd\_get\_x  
    fplot\_core, [57](#)  
surfd\_get\_y  
    fplot\_core, [58](#)  
surfd\_get\_z  
    fplot\_core, [58](#)  
surfd\_set\_data\_1  
    fplot\_core, [58](#)  
surfd\_set\_wireframe  
    fplot\_core, [59](#)  
surfd\_set\_x  
    fplot\_core, [59](#)  
surfd\_set\_y  
    fplot\_core, [59](#)  
surfd\_set\_z  
    fplot\_core, [60](#)  
  
term\_get\_command\_string  
    fplot\_core, [60](#)  
term\_get\_font\_name  
    fplot\_core, [60](#)  
term\_get\_font\_size  
    fplot\_core, [61](#)  
term\_get\_plot\_window\_number  
    fplot\_core, [61](#)  
term\_get\_title  
    fplot\_core, [61](#)  
term\_get\_window\_height  
    fplot\_core, [61](#)  
  
term\_get\_window\_width  
    fplot\_core, [62](#)  
term\_set\_font\_name  
    fplot\_core, [62](#)  
term\_set\_font\_size  
    fplot\_core, [62](#)  
term\_set\_plot\_window\_number  
    fplot\_core, [63](#)  
term\_set\_title  
    fplot\_core, [63](#)  
term\_set\_window\_height  
    fplot\_core, [63](#)  
term\_set\_window\_width  
    fplot\_core, [63](#)  
  
wt\_get\_term\_string  
    fplot\_core, [64](#)  
wxt\_get\_term\_string  
    fplot\_core, [64](#)  
  
xa\_get\_id  
    fplot\_core, [64](#)  
  
y2a\_get\_id  
    fplot\_core, [64](#)  
ya\_get\_id  
    fplot\_core, [65](#)  
  
za\_get\_id  
    fplot\_core, [65](#)