Ostendo

1.0

Generated by Doxygen 1.8.11

Contents

Chapter 1

Namespace Index

1.1 Namespace List

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

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Namespace Documentation

3.1 ostendo Namespace Reference

Core namespace for all Ostendo methods.

Classes

- struct Pos

  Structure to contains position/size data.
- struct Var

  Structure to use as general variable.
- class Window

  Core of everything output to the console.

Enumerations

- enum VarType { INT = 0, DOUBLE = 1, STRING = 2, BOOL = 3 }

  Declares the posible types of the variable structure.
- enum WINDOW_PARTS { WIN_BORDER = 1, WIN_TITLE = 2, WIN_TEXT = 3, WIN_BACKGROUND = 4 }

  Declaration for different components of the window.
- enum ATTRS {
  ALTCHAR = 1, BLINK = 2, BOLD = 3, DIM = 4,
  INVIS = 5, PROTECT = 5, REVERSE = 6, STANDOUT = 7,
  UNDERLINE = 8 }

  Declaration for different output attributes.

Functions

- std::vector< Var > Form (ostendo::Window win, std::vector< Var > fields)
- std::string GetVarStr (Var in_var, int width)
- Var DownArrow (Var in_var, Window win)
- Var UpArrow (Var in_var, Window win)
- Var RightArrow (Var in_var, Window win)
- Var LeftArrow (Var in_var, Window win)
- Var Enter (Var in_var, Window win)
- Var Backspace (Var in_var, Window win)
- Var Add (Var in_var, int ch, Window win)
- std::string Menu (ostendo::Window win, std::vector< std::string > options)
- void InitOstendo (bool color=false, int time_out=-1)

  Initializes Ostendo/Ncurses.
- void TermOstendo ()

  Terminates Ostendo/Ncurses.
- void InitWindow (std::string name)

  Creates a new window.
- void InitWindow (std::string name, int width, int height)

  Creates a new window.
- void InitWindow (std::string name, int width, int height, int x, int y)

  Creates a new window.
- void TermWindow (std::string name)

  Terminates a window.
- void LoadColors ()

  Loads console color pairs.
- std::string TextBox (ostendo::Window win, std::string base="")
- std::string TextBox (ostendo::Window win, std::vector< std::string > base)
- std::vector< std::string > VTextBox (ostendo::Window win, std::string base="")
- std::vector< std::string > VTextBox (ostendo::Window win, std::vector< std::string > base)
- void DisplayLines (ostendo::Window win, std::vector< std::string > lines, int current_line, int &start_line)
- Pos make_pos (int x, int y, int w, int h)

  Creates a new position object.
- Pos make_pos (int x, int y)

  Creates a new position object.

Variables

- Pos std_scr
- std::map< std::string, Window > windows
- int win_index = 0
- bool color_init = false

3.1.1 Detailed Description

Core namespace for all Ostendo methods.

3.1.2 Enumeration Type Documentation

3.1.2.1 enum ostendo::ATTRS

Declaration for different output attributes.

Enumerator

ALTCHAR Uses alternate character set.

BLINK Sets output to blinking (If posible in terminal emulator).

BOLD Sets output to bold.

DIM Sets output to dim colors.

INVIS Sets output text to be invisable.

PROTECT Sets output mode to protected.

REVERSE Sets output to reverse mode.

STANDOUT Sets output to be heighlighted.

UNDERLINE Sets output to be underlined.

3.1.2.2 enum ostendo::VarType

Declares the posible types of the variable structure.

Enumerator

INT Sets the variable type to be integer

DOUBLE Sets the variable type to be doubles

STRING Sets the variable type to be strings

BOOL Sets the variable type to be boolians

3.1.2.3 enum ostendo::WINDOW_PARTS

Declaration for different components of the window.

Enumerator

WIN_BORDER Selects the window border.

WIN_TITLE Selects the window title.

WIN_TEXT Selects the window test.

WIN_BACKGROUND Selects the window background.

3.1.3 Function Documentation

3.1.3.1 ostendo::Var ostendo::Add (Var *in_var,* int *ch,* Window *win*)

Handles form user input of any other key. Provides functionality for different Var types for the user input of any other key.

Parameters

in_var	Var to apply action to.
ch	Character representation of key input.
win	Window to display to.

Returns

Var after action is applied.

See also

Var, Window

3.1.3.2 ostendo::Var ostendo::Backspace (Var *in_var,* Window *win*)

Handles form user input backspace key. Provides functionality for different Var types for the user input of backspace key.

Parameters

in_var	Var to apply action to.
win	Window to display to.

Returns

Var after action is applied.

See also

Var, Window

3.1.3.3 void ostendo::DisplayLines (ostendo::Window *win,* std::vector< std::string > *lines,* int *current_line,* int & *start_line*)

Displays lines from a vector of strings. Displays the lines from a vector of string to a given window, and auto centers the current line, so that it is always on screen.

Parameters

win	Window to display to.
lines	Vector of string to display.
current_line	Current line to center in window.
start_line	line of vector to begin display from.

See also

Window

3.1.3.4 ostendo::Var ostendo::DownArrow (Var *in_var,* Window *win*)

Handles form user input arrow key down. Provides functionality for different Var types for the user input of down arrow key.

Parameters

in_var	Var to apply action to.
win	Window to display to.

Returns

Var after action is applied.

See also

Var, Window

3.1.3.5 ostendo::Var ostendo::Enter (Var *in_var,* Window *win*)

Handles form user input enter key. Provides functionality for different Var types for the user input of enter key.

Parameters

in_var	Var to apply action to.
win	Window to display to.

Returns

Var after action is applied.

See also

Var, Window

3.1.3.6 std::vector< ostendo::Var > ostendo::Form (ostendo::Window *win,* std::vector< Var > *fields*)

Simple form function to get several inputs from the user. This function takes a window, and a set of inputs to request of the user. After the user enters the values for the requested inputs, the values are returned.

Parameters

win	Window to use for displaying inputs.
fields	Vector of Var's for taking in input from the user.

Returns

Vector of Var's filled with user input.

See also

Window, Var

3.1.3.7 std::string ostendo::GetVarStr (Var *in_var,* int *width*)

Gets the string representation of the variable to display. This function reads the Var object, and creates a string to represent the data stored by the object.

Parameters

in_var	Var to interpret.
width	Maximum width for return string.

Returns

String representation of in_var.

See also

Var.

3.1.3.8 void ostendo::InitOstendo (bool *color* = `false`, int *time_out* = `-1`)

Initializes Ostendo/Ncurses.

Initializes ncurses with default settings, and enables keypad input, and reads std_scr data. Optionaly enables color, disables cursor, and sets input timeout time.

Parameters

color	Determines if colors should be enabled.
cursor	Determines if cursor should be disabled.
time_out	Sets the time out time for intput. -1 for no time out.

See also

std_scr, windows, color_init, TermOstendo()

3.1.3.9 void ostendo::InitWindow (std::string *name*)

Creates a new window.

Creates a new window, and adds it to windows. The window has the same dimensions of std_scr.

Parameters

name	The name of the window for the key in windows map.

See also

windows, std_scr

3.1.3.10 void ostendo::InitWindow (std::string *name,* int *width,* int *height*)

Creates a new window.

Creates a new window, and adds it to windows. The window is positioned at 0,0 with given width and height.

Parameters

name	The name of the window for the key in windows map.
width	The width of the window in characters.
height	The height of the window in characters.

See also

windows, std_scr

3.1.3.11 void ostendo::InitWindow (std::string *name,* int *width,* int *height,* int *x,* int *y*)

Creates a new window.

Creates a new window, and adds it to windows. The window is at a given position with a given width and height.

Parameters

name	The name of the window for the key in windows map.
width	The width of the window in characters.
height	The height of the window in characters.
x	The x position of the window in characters from the left.
y	the y position of the window in characters from the top.

See also

windows, std_scr

3.1.3.12 ostendo::Var ostendo::LeftArrow (Var *in_var,* Window *win*)

Handles form user input arrow key left. Provides functionality for different Var types for the user input of left arrow key.

Parameters

in_var	Var to apply action to.
win	Window to display to.

Returns

Var after action is applied.

See also

Var, Window

3.1.3.13 void ostendo::LoadColors ( )

Loads console color pairs.

Loads every combination of the console color pairs, if colors are initialized.

See also

color_init

3.1.3.14 Pos ostendo::make_pos (int *x,* int *y,* int *w,* int *h*) [inline]

Creates a new position object.

Creates position object given the four terms.

Parameters

x	X term for position.
y	Y term for position.
w	Width term for position.
h	Height term for position.

Returns

Pos New position structure from given values.

3.1.3.15 Pos ostendo::make_pos (int *x*, int *y*) [inline]

Creates a new position object.

Creates a position object given two terms.

Parameters

x	X term for position.
y	Y term for position.

Returns

Pos New position structure from given values.

3.1.3.16 std::string ostendo::Menu (ostendo::Window *win*, std::vector< std::string > *options*)

Menu function for selecting an option from a set. This function takes a set of options, and a window, then provides a simple user interface for selecting one of the provided options. This selected option is then returned.

Parameters

win	Window to use to display all options.
options	Options that the user must select from.

Returns

String from selected option of Options.

See also

Window

3.1.3.17 ostendo::Var ostendo::RightArrow (Var *in_var*, Window *win*)

Handles form user input arrow key right. Provides functionality for different Var types for the user input of right arrow key.

Parameters

in_var	Var to apply action to.
win	Window to display to.

Returns

Var after action is applied.

See also

Var, Window

3.1.3.18 void ostendo::TermOstendo ( )

Terminates Ostendo/Ncurses.

Terminates ncurses, and returns the console to standard settings.

See also

InitOstendo()

3.1.3.19 void ostendo::TermWindow (std::string *name*)

Terminates a window.

Termainates a specified window from the windows map.

Parameters

name	The name of the window to be deleted.

See also

windows

3.1.3.20 std::string ostendo::TextBox (ostendo::Window *win,* std::string *base* = " ")

Function for basic extended text entry from user. This function alows the user to write text to a text box, and when compleated, the writen text is formated into a string.

Parameters

win	window to display to.
base	Starting string to display in text box.

Returns

String of what user entered.

See also

Window

3.1.3.21 std::string ostendo::TextBox (ostendo::Window *win*, std::vector< std::string > *base*)

Function for basic extended text entry from user. This function alows the user to write text to a text box, and when compleated, the writen text is formated into a string.

Parameters

win	window to display to.
base	Starting vector of strings representing different lines of text.

Returns

String of what user entered.

See also

Window

3.1.3.22 ostendo::Var ostendo::UpArrow (Var *in_var*, Window *win*)

Handles form user input arrow key up. Provides functionality for different Var types for the user input of up arrow key.

Parameters

in_var	Var to apply action to.
win	Window to display to.

Returns

Var after action is applied.

See also

Var, Window

3.1.3.23 std::vector< std::string > ostendo::VTextBox (ostendo::Window *win*, std::string *base* = "")

Function for basic extended text entry from user. This function alows the user to write text to a text box, and when compleated, the writen text is formated into a a vector of strings, each representing a lines of text.

Parameters

win	window to display to.
base	Starting string to display in text box.

Returns

Vector of strings representing the different lines of input from the user.

See also

Window

3.1.3.24 std::vector< std::string > ostendo::VTextBox (ostendo::Window *win,* std::vector< std::string > *base*)

Function for basic extended text entry from user. This function alows the user to write text to a text box, and when compleated, the writen text is formated into a a vector of strings, each representing a lines of text.

Parameters

win	window to display to.
base	Starting vector of stirngs representing different lines of text.

Returns

Vector of strings representing the different lines of input from the user.

See also

Window

3.1.4 Variable Documentation

3.1.4.1 bool ostendo::color_init = false

Boolian to check if colors have been initialized.

3.1.4.2 Pos ostendo::std_scr

Width and height data for full window, measured at startup.

3.1.4.3 int ostendo::win_index = 0

Current window index, from windows map.

3.1.4.4 std::map< std::string, Window > ostendo::windows

Optional map for windows, to link windows with given name.

Chapter 4

Class Documentation

4.1 ostendo::Pos Struct Reference

Structure to contains position/size data.

`#include <types.hpp>`

Public Member Functions

- Pos ()

  Creates new position.
- Pos (int value_x, int value_y)

  Creates new position.
- Pos (int value_x, int value_y, int value_w, int value_h)

  Creates new position.

Public Attributes

- int x
- int y
- int w
- int h

4.1.1 Detailed Description

Structure to contains position/size data.

Contains the x, y positional data, and the width, height size data.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 ostendo::Pos::Pos ( ) `[inline]`

Creates new position.

Creates position with zero for all terms

4.1.2.2 ostendo::Pos::Pos (int *value_x,* int *value_y*) `[inline]`

Creates new position.

Creates position with set x and y position, and width and height are set to zero

Parameters

value←↩ _x	Value for the X position.
value←↩ _y	Value for the y position.

4.1.2.3 ostendo::Pos::Pos (int *value_x,* int *value_y,* int *value_w,* int *value_h*) [inline]

Creates new position.

Creates position data with set x, y, width, and height data

Parameters

value←↩ _x	Value for the x position.
value←↩ _y	Value for the y position.
value←↩ _w	Value for the width.
value←↩ _h	Value for the height.

4.1.3 Member Data Documentation

4.1.3.1 int ostendo::Pos::h

Height term for the position data.

4.1.3.2 int ostendo::Pos::w

Width term for the position data.

4.1.3.3 int ostendo::Pos::x

X term of the position data.

4.1.3.4 int ostendo::Pos::y

Y term of the position data.

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

- ostendo_files/types.hpp

4.2 ostendo::Var Struct Reference

Structure to use as general variable.

```
#include <types.hpp>
```

Public Member Functions

- Var ()

  Creates new Var.
- Var (int value)

  Creates new Var.
- Var (double value)

  Creates new Var.
- Var (std::string value)

  Creates new Var.
- Var (bool value)

  Creates new Var.
- Var (std::string str_name, int value)

  Creates new Var.
- Var (std::string str_name, double value)

  Creates new Var.
- Var (std::string str_name, std::string value)

  Creates new Var.
- Var (std::string str_name, bool value)

  Creates new Var.
- Var (std::string str_name, std::vector< int > set, int value=0)

  Creates new Var set.
- Var (std::string str_name, std::vector< double > set, int value=0)

  Creates new Var set.
- Var (std::string str_name, std::vector< std::string > set, int value=0)

  Creates new Var set.
- void operator= (int value)

  Sets the value of the variable.
- void operator= (double value)

  Sets the value of the variable.
- void operator= (std::string value)

  Sets the value of the variable.
- void operator= (bool value)

  Sets the value of the variable.
- void SetSet (std::vector< int > set)

  Sets the set of the variable.
- void SetSet (std::vector< double > set)

  Sets the set of the variable.
- void SetSet (std::vector< std::string > set)

  Sets the set of the variable.
- int Int ()

  Returns integer value of the variable.
- double Double ()

  Returns the double value of the varaible.

- std::string String ()

  Returns the string value of the variable.
- bool Bool ()

  Returns the boolian value of the variable.
- std::string Name ()

  Returns the name of the variable.

Public Attributes

- bool has_set
- std::string name
- int var_type
- int int_value
- double double_value
- std::string str_value
- bool bool_value
- std::vector< int > int_set
- std::vector< double > double_set
- std::vector< std::string > str_set

4.2.1 Detailed Description

Structure to use as general variable.

Structure that can act as a variable for int, double, string, or bool

4.2.2 Constructor & Destructor Documentation

4.2.2.1 ostendo::Var::Var ( ) [inline]

Creates new Var.

Creates the default variable, with standard values for all terms.

4.2.2.2 ostendo::Var::Var (int *value*) [inline]

Creates new Var.

Creates new integer variable.

Parameters

value	Integer value for variable.

4.2.2.3 ostendo::Var::Var (double *value*) [inline]

Creates new Var.

Creates new double variable.

Parameters

value	Double value for variable.

4.2.2.4 ostendo::Var::Var (std::string *value*) [inline]

Creates new Var.

Creates new string variable.

Parameters

value	String value for variable.

4.2.2.5 ostendo::Var::Var (bool *value*) [inline]

Creates new Var.

Creates new boolian variable.

Parameters

value	Boolian value for variable.

4.2.2.6 ostendo::Var::Var (std::string *str_name,* int *value*) [inline]

Creates new Var.

Creates a new integer variable.

Parameters

str_name	Name of variable.
value	Integer value for variable.

4.2.2.7 ostendo::Var::Var (std::string *str_name,* double *value*) [inline]

Creates new Var.

Creates a new double variable.

Parameters

str_name	Name of variable.
value	Double value for variable.

4.2.2.8 ostendo::Var::Var (std::string *str_name,* std::string *value*) [inline]

Creates new Var.

Creates a new string variable.

Parameters

str_name	Name of variable.
value	String value for variable.

4.2.2.9 ostendo::Var::Var (std::string *str_name,* bool *value*) [inline]

Creates new Var.

Creates a new boolian variable.

Parameters

str_name	Name of variable.
value	Boolian value for variable.

4.2.2.10 ostendo::Var::Var (std::string *str_name,* std::vector< int > *set,* int *value* = 0) [inline]

Creates new Var set.

Creates new variable set of integers with given default.

Parameters

str_name	Name of variable.
set	Set of integer options.
value	Default option from the set.

4.2.2.11 ostendo::Var::Var (std::string *str_name,* std::vector< double > *set,* int *value* = 0) [inline]

Creates new Var set.

Creates new variable set of doubles with given default.

Parameters

str_name	Name of variable.
set	Set of double options.
value	Default option from the set.

4.2.2.12 ostendo::Var::Var (std::string *str_name*, std::vector< std::string > *set*, int *value* = 0) [inline]

Creates new Var set.

Creates new variable set of strings with given default.

Parameters

str_name	Name of variable.
set	Set of strings options.
value	Default option from the set.

4.2.3 Member Function Documentation

4.2.3.1 void ostendo::Var::operator= (int *value*) [inline]

Sets the value of the variable.

Sets the variable to the given integer value.

Parameters

value	Value to set variable to.

4.2.3.2 void ostendo::Var::operator= (double *value*) [inline]

Sets the value of the variable.

Sets the variable to the given double value.

Parameters

value	Value to set variable to.

4.2.3.3 void ostendo::Var::operator= (std::string *value*) [inline]

Sets the value of the variable.

Sets the variable to the given string value.

Parameters

value	Value to set variable to.

4.2.3.4 void ostendo::Var::operator= (bool *value*) `[inline]`

Sets the value of the variable.

Sets the variable to the given boolian value.

Parameters

value	Value to set variable to.

4.2.3.5 void ostendo::Var::SetSet (std::vector< int > *set*) `[inline]`

Sets the set of the variable.

Set the variable set to a set of integers.

Parameters

set	Set of integers to use in the variable.

4.2.3.6 void ostendo::Var::SetSet (std::vector< double > *set*) `[inline]`

Sets the set of the variable.

Set the variable set to a set of doubles.

Parameters

set	Set of doubles to use in the variable.

4.2.3.7 void ostendo::Var::SetSet (std::vector< std::string > *set*) `[inline]`

Sets the set of the variable.

Set the variable set to a set of strings.

Parameters

set	Set of strings to use in the variable.

4.2.4 Member Data Documentation

4.2.4.1 bool ostendo::Var::bool_value

Boolian value.

4.2.4.2 std::vector<double> ostendo::Var::double_set

Set of double options.

4.2.4.3 double ostendo::Var::double_value

Double value.

4.2.4.4 bool ostendo::Var::has_set

Boolian determining if variable is a set.

4.2.4.5 std::vector<int> ostendo::Var::int_set

Set of integer options.

4.2.4.6 int ostendo::Var::int_value

Integer value.

4.2.4.7 std::string ostendo::Var::name

Name of variable.

4.2.4.8 std::vector<std::string> ostendo::Var::str_set

Set of string options.

4.2.4.9 std::string ostendo::Var::str_value

String value.

4.2.4.10 int ostendo::Var::var_type

Variable type setting from VarTypes.

See also

VarType

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

- ostendo_files/types.hpp

4.3 ostendo::Window Class Reference

Core of everything output to the console.

```
#include <window.hpp>
```

Public Member Functions

- Window ()

  Creates a new window.
- Window (int width, int height)

  Creates a new window.
- Window (double width, double height)

  Creates a new window.
- Window (int width, int height, int pos_x, int pos_y)

  Creates a new window.
- Window (int width, int height, double pos_x, double pos_y)

  Creates a new window.
- Window (double width, double height, double pos_x, double pos_y)

  Creates a new window.
- Window (double width, double height, int pos_x, int pos_y)

  Creates a new window.
- Window (Pos pos)

  Creates a new window.
- Window (const Window &win)

  Copies a window.
- void DelWin ()

  Deletes window.
- void NewWindow ()

  Deletes window and creates a new window.
- void NewWindow (int width, int height)

  Deletes window and creates a new window.
- void NewWindow (double width, double height)

  Deletes window and creates a new window.
- void NewWindow (int width, int height, int pos_x, int pos_y)

  Deletes window and creates a new window.

- void NewWindow (int width, int height, double pos_x, double pos_y)

  Deletes window and creates a new window.
- void NewWindow (double width, double height, double pos_x, double pos_y)

  Deletes window and creates a new window..
- void NewWindow (double width, double height, int pos_x, int pos_y)

  Deletes window and creates a new window.
- void Update ()

  Updates window display.
- void ToggleBorder ()

  Toggles drawing window border.
- void ToggleScroll ()

  Toggles window scrolling.
- void ToggleTitle (std::string setting="")

  Toggles window title.
- void SetAutoUpdate (bool setting=true)

  Sets window auto updating.
- void ColorOn (int attrs)

  Enables a color for output.
- void ColorOff (int attrs)

  Disables a color for output.
- void AttrOn (int attrs)

  Enables an attribute for output.
- void AttrOff (int attrs)

  Disables an attribute for output.
- void SetColor (int color, int value)

  Sets display color of window part.
- void Clear (bool all=false)

  Clears window of outputs.
- void ClearLine (int line=-1)

  Clears the current line of window.
- int Print (std::string str,...)

  Print output to window.
- void SetCurs (int y=-1, int x=-1)

  Sets cursor position in window.
- WINDOW ∗ operator() ()

  Gets ncurses window pointer.

Public Attributes

- Pos window_space

4.3.1 Detailed Description

Core of everything output to the console.

The Window class is the base of everything that is output to the console. Everything output must be part of a window, and the windows controll all outputs, and output attributes.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 ostendo::Window::Window ( )

Creates a new window.

Creates a new window with the same base setting of std_scr.

See also

std_scr

4.3.2.2 ostendo::Window::Window (int *width*, int *height*)

Creates a new window.

Creates a new window with given width and height, at 0,0.

Parameters

width	Width of window.
height	Height of window.

4.3.2.3 ostendo::Window::Window (double *width*, double *height*)

Creates a new window.

Creates a new window with width and height as a percentage of std_scr.

Parameters

width	Percentage of width.
height	Percentage of height.

See also

std_scr

4.3.2.4 ostendo::Window::Window (int *width*, int *height*, int *pos_x*, int *pos_y*)

Creates a new window.

Creates a new window with given width and height, at given position.

Parameters

width	Width of window.

Parameters

height	Height of window.
pos↩ _x	X position of window.
pos↩ _y	Y position of window.

4.3.2.5 ostendo::Window::Window (int *width*, int *height*, double *pos_x*, double *pos_y*)

Creates a new window.

Creates a new window with given width and height, and at x and y of percentage of std_scr.

Parameters

width	Width of window.
height	Height of window.
pos↩ _x	Percentage of width.
pos↩ _y	Percentage of height.

See also

std_scr

4.3.2.6 ostendo::Window::Window (double *width*, double *height*, double *pos_x*, double *pos_y*)

Creates a new window.

Creates a new window with width and height as a percentage of std_scr, and at x and y of percentage of std_scr.

Parameters

width	Percentace of width.
height	Percentage of height.
pos↩ _x	Percentage of width.
pos↩ _y	Percentage of height.

See also

std_scr

4.3.2.7 ostendo::Window::Window (double *width*, double *height*, int *pos_x*, int *pos_y*)

Creates a new window.

Creates a new window with given width and height, and at x and y.

Parameters

width	Percentage of width.
height	Percentage of height.
pos↩ _x	X position of window.
pos↩ _y	Y position of window.

See also

std_scr

4.3.2.8 ostendo::Window::Window (Pos *pos*)

Creates a new window.

Creates a window with the width, height, and position given by Position object.

Parameters

pos	Position definition of window.

See also

Pos

4.3.2.9 ostendo::Window::Window (const Window & *win*)

Copies a window.

Creates a copy of a given window.

Parameters

win	Window to create a copy of.

4.3.3 Member Function Documentation

4.3.3.1 void ostendo::Window::AttrOff (int *attrs*)

Disables an attribute for output.

Disables previously enabled output attribute.

Parameters

attrs	Attribute value from Attrs.

See also

Attrs

4.3.3.2 void ostendo::Window::AttrOn (int *attrs*)

Enables an attribute for output.

Enables output attribute.

Parameters

attrs	Attribute value from Attrs.

See also

Attrs

4.3.3.3 void ostendo::Window::Clear (bool *all* = `false`)

Clears window of outputs.

Clears window of all ouputs.

Parameters

all	If true, then clears window of border and title aswell.

4.3.3.4 void ostendo::Window::ClearLine (int *line* = `-1`)

Clears the current line of window.

Clears the current line of cursor y, and sets cursor x to begining of the line.

Parameters

line	Specify line to clear.

4.3.3.5 void ostendo::Window::ColorOff (int *attrs*)

Disables a color for output.

Disables previously enabled output color.

Parameters

attrs	Color index represented by a two digit number.

4.3.3.6 void ostendo::Window::ColorOn (int *attrs*)

Enables a color for output.

Enables output color.

Parameters

attrs	Color index represented by a two digit number.

4.3.3.7 void ostendo::Window::NewWindow ( )

Deletes window and creates a new window.

Creates a new window with the same base setting of std_scr.

See also

std_scr

4.3.3.8 void ostendo::Window::NewWindow (int *width,* int *height*)

Deletes window and creates a new window.

Creates a new window with given width and height, at 0,0.

Parameters

width	Width of window.
height	Height of window.

4.3.3.9 void ostendo::Window::NewWindow (double *width,* double *height*)

Deletes window and creates a new window.

Creates a new window with width and height as a percentage of std_scr.

Parameters

width	Percentage of width.
height	Percentage of height.

See also

std_scr

4.3.3.10 void ostendo::Window::NewWindow (int *width,* int *height,* int *pos_x,* int *pos_y*)

Deletes window and creates a new window.

Creates a new window with given width and height, at given position.

Parameters

width	Width of window.
height	Height of window.
pos↩_x	X position of window.
pos↩_y	Y position of window.

4.3.3.11 void ostendo::Window::NewWindow (int *width,* int *height,* double *pos_x,* double *pos_y*)

Deletes window and creates a new window.

Creates a new window with given width and height, and at x and y of percentage of std_scr.

Parameters

width	Width of window.
height	Height of window.
pos↩_x	Percentage of width.
pos↩_y	Percentage of height.

See also

std_scr

4.3.3.12 void ostendo::Window::NewWindow (double *width,* double *height,* double *pos_x,* double *pos_y*)

Deletes window and creates a new window..

Creates a new window with width and height as a percentage of std_scr, and at x and y of percentage of std_scr.

Parameters

width	Percentace of width.
height	Percentage of height.
pos↩_x	Percentage of width.
pos↩_y	Percentage of height.

See also

std_scr

4.3.3.13 void ostendo::Window::NewWindow (double *width*, double *height*, int *pos_x*, int *pos_y*)

Deletes window and creates a new window.

Creates a new window with given width and height, and at x and y.

Parameters

width	Percentage of width.
height	Percentage of height.
pos↩_x	X position of window.
pos↩_y	Y position of window.

See also

std_scr

4.3.3.14 WINDOW∗ ostendo::Window::operator() ( ) [inline]

Gets ncurses window pointer.

Gets the ncurses window pointer for the current window.

Returns

pointer to ncurses window.

4.3.3.15 int ostendo::Window::Print (std::string *str*, ...)

Print output to window.

Main output function for ostendo. str is a formated string such as for printf, with additional arguments as required. Accepts additional excape sequenced initialized with a '#'. '#o__' will turn on specified attribute, while '#f__' will turn off the specified attribute. Attributes can be two digit numbers (0-7)(0-7), where the first number is the forground color, and the second number is the backround color. These attributes can also be any of the following: ac(alternate character set), bl(blink), bo(bold), di(dim), in(invisible), pr(protected), re(reverse), st(standout), un(underlined).

Parameters

str	Format string.
...	Additional arguments as required.

Returns

-1 if window has not been initialized, otherwise 0.

4.3.3.16 void ostendo::Window::SetAutoUpdate (bool *setting* = `true`)

Sets window auto updating.

Sets window auto updating to specified value.

Parameters

setting	Value to set auto updating to.

4.3.3.17 void ostendo::Window::SetColor (int *color,* int *value*)

Sets display color of window part.

Sets the color of compnents of the window.

Parameters

color	Color represented by two digit number.
value	Window component to color.

4.3.3.18 void ostendo::Window::SetCurs (int *y* = `-1`, int *x* = `-1`)

Sets cursor position in window.

Sets the position of the output cursor for the window.

Parameters

x	X position of the cursor.
y	Y position of the cursor.

4.3.3.19 void ostendo::Window::ToggleBorder ( )

Toggles drawing window border.

Toggles wether the window border should be drawn.

4.3.3.20 void ostendo::Window::ToggleScroll ( )

Toggles window scrolling.

Toggles if the window should handle last line output by scrolling.

4.3.3.21 void ostendo::Window::ToggleTitle (std::string *setting* = " ")

Toggles window title.

Toggles wether the window title should be drawn.

Parameters

setting	String to set as the new window title.

4.3.3.22 void ostendo::Window::Update ( )

Updates window display.

Updates the window display, to show recently printed outputs. Must be called for display to be visible.

4.3.4 Member Data Documentation

4.3.4.1 Pos ostendo::Window::window_space

Window space avalible for output (Excluding borders/title).

The documentation for this class was generated from the following files:

- ostendo_files/window.hpp
- ostendo_files/window.cpp

Index