

Enlightened Widget Library Reference Manual

Generated by Doxygen 1.5.1

Wed Mar 28 00:01:51 2007

Contents

1	The Enlightened Widget Library	1
1.1	Introduction	1
1.2	The Object Model	1
1.3	Callbacks	2
1.4	Example Application Walk-through	2
1.5	Conclusion	4
2	Enlightened Widget Library Module Index	5
2.1	Enlightened Widget Library Modules	5
3	Enlightened Widget Library Data Structure Index	7
3.1	Enlightened Widget Library Data Structures	7
4	Enlightened Widget Library File Index	13
4.1	Enlightened Widget Library File List	13
5	Enlightened Widget Library Page Index	19
5.1	Enlightened Widget Library Related Pages	19
6	Enlightened Widget Library Module Documentation	21
6.1	Ewl_Attach: A mechanism to attach data to widgets.	21
6.2	Ewl_Border: A container with a border and label	26
6.3	Ewl_Box: The Box Layout Container.	30
6.4	Ewl_Button: The Basic Button	35
6.5	Ewl_Calendar: The EWL Calendar Widget	41
6.6	Ewl_Callback: The Callback Mechanisms	44
6.7	Ewl_Cell: The Cell Container, Layout for a Single Widget	51
6.8	Ewl_Check: The Simple Check for a Checkbutton	53
6.9	Ewl_Checkbutton: An Ewl_Checkbutton with Label	56
6.10	Ewl_Colordialog: A Simple Colour Dialog widget	59

6.11 Ewl_Colorpicker: The colour picking widget	65
6.12 Ewl_Combo: A Simple Combo Box	71
6.13 Ewl_Config: Functions for Manipulating Configuration Data	78
6.14 Ewl_Container: Widgets Holding Other Widgets	84
6.15 Ewl_Context_Menu: A context menu	100
6.16 Ewl_Cursor: Container to create custom mouse cursors	104
6.17 Ewl_Datepicker: The EWL Datepicker widget	106
6.18 Ewl_Dialog: A dialog window	108
6.19 Ewl_Dnd: The files containing DND functions	112
6.20 Ewl_Embed: A Container for Displaying on an canvas	118
6.21 Ewl_Entry: A text entry widget	136
6.22 Ewl_Enums: Various Flags and Enumerations used in EWL	144
6.23 Ewl_Events: Lower Level Event Handlers	166
6.24 Ewl_Expansion: The Expansion node for Tree2	170
6.25 Ewl_Filedialog: A Dialog For Picking Files	173
6.26 Ewl_Filelist: The base widget for the filelists	180
6.27 Ewl_Filelist_Column: An expanding column view	190
6.28 Ewl_Filelist_Icon: A icon file view	193
6.29 Ewl_Filelist_List: A listview of the files	196
6.30 Ewl_Filelist_Tree: A treeview of the files	199
6.31 Ewl_Filepicker: Allows selecting files from a list	202
6.32 Ewl_Floater: A Floating Container	210
6.33 Ewl_Freebox: A Freelayout widget	213
6.34 Ewl_Grid The ewl grid widget	218
6.35 Ewl_Highlight: A highlight widget	229
6.36 Ewl_Histogram: A Simple Histogram widget	232
6.37 Ewl_Icon: An icon widget	236
6.38 Ewl_Icon_Theme: The icon theme code	245
6.39 Ewl_Image: An Image Display Widget	276
6.40 Ewl_Label: A fully-themable text label	287
6.41 Ewl_List: A list widget	290
6.42 Ewl_Macros: Useful Macros Used Internally and Available Externally	292
6.43 Ewl_Media: A Multi media widget	295
6.44 Ewl_Menu: A Simple Windowed Menu	302
6.45 Ewl_Menu_Item: The basic menu item	306
6.46 Ewl_Menubar: A menu bar widget	308

6.47 Ewl_Misc: Miscellaneous Utility Functions	312
6.48 Ewl_Model: A data model	315
6.49 Ewl_MVC: The base model-view-controller framework	326
6.50 Ewl_Notebook: A notebook widget	336
6.51 Ewl_Object: Base object. Provides functionality available to all widgets.	342
6.52 Ewl_Overlay: A Container for Displaying on a canvas	372
6.53 Ewl_Paned: A paned widget	374
6.54 Ewl_Password: A Single Line Text Password Widget	379
6.55 Ewl_Popup: A override window for the convenience use	383
6.56 Ewl_Progressbar: A progressbar from a value	388
6.57 Ewl_Radiobutton: A Radio Button Widget and Grouping System	392
6.58 Ewl_Range: A Value Selector from a Range	395
6.59 Ewl_Row: The Row widget	401
6.60 Ewl_Scrollbar: A Simple Scrollbar Widget	404
6.61 Ewl_Scrollpane: A Scrollable Viewing Area	409
6.62 Ewl_Seeker: A Value Selector from a Range	415
6.63 Ewl_Separator: A Visual Separator Between Widgets	419
6.64 Ewl_Shadow: A container with a shadow/border	422
6.65 Ewl_Spacer: A Widget to Add Space Between Other Widgets	424
6.66 Ewl_Spectrum: The colour spectrum widget	426
6.67 Ewl_Spinner: A Numerical Value Entry	432
6.68 Ewl_Statusbar: A status bar widget	436
6.69 Ewl_Stock: The Stock	442
6.70 Ewl_Table: The Table Layout Container.	445
6.71 Ewl_Text: A text layout widget	452
6.72 Ewl_Theme: Methods for Accessing and Modifying Theme Data	486
6.73 Ewl_Toolbar: A toolbar widget	490
6.74 Ewl_Tree: A Widget for List or Tree Layout	493
6.75 Ewl_Tree2: A Widget for List or Tree Layout	505
6.76 Ewl_Tree2_View: A tree view	527
6.77 Ewl_Tree2_View_Plain: The plain tree view	529
6.78 Ewl_Tree2_View_Scrolled The scrolled tree view	531
6.79 Ewl_View: A data view	534
6.80 Ewl_Widget: The Parent Widget Class Common to All Widgets	539
6.81 Ewl_Window: A Container for Displaying in a New Window	563

7 Enlightened Widget Library Data Structure Documentation

577

7.1	Combo_Test_Data Struct Reference	577
7.2	Control Struct Reference	578
7.3	Ewl_Attach Struct Reference	579
7.4	Ewl_Attach_Dnd Struct Reference	580
7.5	Ewl_Attach_List Struct Reference	581
7.6	Ewl_Attach_Tooltip Struct Reference	582
7.7	Ewl_Border Struct Reference	584
7.8	Ewl_Box Struct Reference	586
7.9	Ewl_Box_Orientation Struct Reference	588
7.10	Ewl_Button Struct Reference	589
7.11	Ewl_Calendar Struct Reference	591
7.12	Ewl_Callback Struct Reference	593
7.13	Ewl_Callback_Chain Struct Reference	594
7.14	Ewl_Callback_Custom Struct Reference	595
7.15	Ewl_Cell Struct Reference	596
7.16	Ewl_Check Struct Reference	597
7.17	Ewl_Checkbutton Struct Reference	598
7.18	Ewl_Color_Set Struct Reference	600
7.19	Ewl_Colordialog Struct Reference	601
7.20	Ewl_Colorpicker Struct Reference	603
7.21	Ewl_Colorpicker_Radiobutton Struct Reference	607
7.22	Ewl_Colorpicker_Spinner Struct Reference	609
7.23	Ewl_Combo Struct Reference	610
7.24	Ewl_Combo_Cell Struct Reference	612
7.25	Ewl_Config Struct Reference	614
7.26	Ewl_Config_Cache Struct Reference	615
7.27	Ewl_Configure_Queue Struct Reference	617
7.28	Ewl_Container Struct Reference	618
7.29	Ewl_Context_Menu Struct Reference	620
7.30	Ewl_Cursor Struct Reference	622
7.31	Ewl_Datepicker Struct Reference	624
7.32	Ewl_Dialog Struct Reference	626
7.33	Ewl_Dnd_Types Struct Reference	628
7.34	Ewl_Embed Struct Reference	629
7.35	Ewl_Engine Struct Reference	633
7.36	Ewl_Engine_Info Struct Reference	634

7.37 Ewl_Entry Struct Reference	635
7.38 Ewl_Entry_Cursor Struct Reference	637
7.39 Ewl_Event_Action_Response Struct Reference	639
7.40 Ewl_Event_Dnd_Data_Received Struct Reference	640
7.41 Ewl_Event_Dnd_Data_Request Struct Reference	641
7.42 Ewl_Event_Dnd_Drop Struct Reference	642
7.43 Ewl_Event_Dnd_Position Struct Reference	643
7.44 Ewl_Event_Key Struct Reference	644
7.45 Ewl_Event_Key_Down Struct Reference	645
7.46 Ewl_Event_Key_Up Struct Reference	646
7.47 Ewl_Event_Mouse Struct Reference	647
7.48 Ewl_Event_Mouse_Down Struct Reference	648
7.49 Ewl_Event_Mouse_In Struct Reference	649
7.50 Ewl_Event_Mouse_Move Struct Reference	650
7.51 Ewl_Event_Mouse_Out Struct Reference	651
7.52 Ewl_Event_Mouse_Up Struct Reference	652
7.53 Ewl_Event_Mouse_Wheel Struct Reference	653
7.54 Ewl_Event_Window_Expose Struct Reference	654
7.55 Ewl_Expansion Struct Reference	655
7.56 Ewl_Filedialog Struct Reference	656
7.57 Ewl_Filelist Struct Reference	658
7.58 Ewl_Filelist_Column Struct Reference	661
7.59 Ewl_Filelist_Column_Data Struct Reference	663
7.60 Ewl_Filelist_Icon Struct Reference	664
7.61 Ewl_Filelist_List Struct Reference	665
7.62 Ewl_Filelist_Tree Struct Reference	666
7.63 Ewl_Filelist_Tree_Data Struct Reference	667
7.64 Ewl_Filepicker Struct Reference	668
7.65 Ewl_Filepicker_Filter Struct Reference	670
7.66 Ewl_Floater Struct Reference	671
7.67 Ewl_Freebox Struct Reference	673
7.68 Ewl_Grid Struct Reference	675
7.69 Ewl_Grid_Child Struct Reference	677
7.70 Ewl_Grid_Info Struct Reference	678
7.71 Ewl_Highlight Struct Reference	680
7.72 Ewl_Histogram Struct Reference	681

7.73 Ewl_Icon Struct Reference	683
7.74 Ewl_Image Struct Reference	686
7.75 Ewl_Image_Thumbnail Struct Reference	689
7.76 Ewl_IO_Manager_Plugin Struct Reference	690
7.77 Ewl_Label Struct Reference	692
7.78 Ewl_List Struct Reference	693
7.79 Ewl_Media Struct Reference	694
7.80 Ewl_Menu Struct Reference	696
7.81 Ewl_Menu_Info Struct Reference	698
7.82 Ewl_Menu_Item Struct Reference	699
7.83 Ewl_Menubar Struct Reference	701
7.84 Ewl_Menubar_Info Struct Reference	702
7.85 Ewl_Model Struct Reference	703
7.86 Ewl_MVC Struct Reference	705
7.87 Ewl_Notebook Struct Reference	707
7.88 Ewl_Object Struct Reference	709
7.89 Ewl_Overlay Struct Reference	712
7.90 Ewl_Pair Struct Reference	713
7.91 Ewl_Pair_List Struct Reference	714
7.92 Ewl_Paned Struct Reference	715
7.93 Ewl_Paned_Grabber Struct Reference	717
7.94 Ewl_Paned_Layout Struct Reference	718
7.95 Ewl_Password Struct Reference	719
7.96 Ewl_Popup Struct Reference	721
7.97 Ewl_Progressbar Struct Reference	723
7.98 Ewl_Radiobutton Struct Reference	725
7.99 Ewl_Range Struct Reference	727
7.100Ewl_Row Struct Reference	729
7.101Ewl_Scrollbar Struct Reference	730
7.102Ewl_Scrollpane Struct Reference	732
7.103Ewl_Seeker Struct Reference	734
7.104Ewl_Selection Struct Reference	736
7.105Ewl_Selection_Idx Struct Reference	737
7.106Ewl_Selection_Range Struct Reference	738
7.107Ewl_Separator Struct Reference	740
7.108Ewl_Shadow Struct Reference	741

7.109Ewl_Spacer Struct Reference	742
7.110Ewl_Spectrum Struct Reference	743
7.111Ewl_Spinner Struct Reference	746
7.112Ewl_Statusbar Struct Reference	748
7.113Ewl_Stock Struct Reference	750
7.114Ewl_Stock_Funcs Struct Reference	752
7.115Ewl_Table Struct Reference	753
7.116Ewl_Test Struct Reference	755
7.117Ewl_Text Struct Reference	757
7.118Ewl_Text_Context Struct Reference	761
7.119Ewl_Text_Fmt Struct Reference	764
7.120Ewl_Text_Trigger Struct Reference	765
7.121Ewl_Text_Trigger_Area Struct Reference	767
7.122Ewl_Toolbar Struct Reference	768
7.123Ewl_Tree Struct Reference	769
7.124Ewl_Tree2 Struct Reference	771
7.125Ewl_Tree2_Node Struct Reference	774
7.126Ewl_Tree2_View Struct Reference	776
7.127Ewl_Tree2_View_Plain Struct Reference	778
7.128Ewl_Tree2_View_Scrolled Struct Reference	779
7.129Ewl_Tree_Node Struct Reference	781
7.130Ewl_Unit_Test Struct Reference	783
7.131Ewl_View Struct Reference	784
7.132Ewl_Widget Struct Reference	785
7.133Ewl_Window Struct Reference	788
7.134Freebox_Test Struct Reference	790
7.135List_Test_Data Struct Reference	791
7.136List_Test_Row_Data Struct Reference	792
7.137Tree2_Test_Data Struct Reference	793
7.138Tree2_Test_Row_Data Struct Reference	794
8 Enlightened Widget Library File Documentation	795
8.1 bin/ewl_embed_test.c File Reference	795
8.2 bin/ewl_simple_test.c File Reference	797
8.3 bin/Ewl_Test.h File Reference	798
8.4 bin/ewl_test_private.h File Reference	801
8.5 bin/main.c File Reference	804

8.6	bin/tests/dnd_snoop/ewl_dnd_snoop.c File Reference	806
8.7	bin/tests/engine/ewl_engine.c File Reference	808
8.8	bin/tests/fullscreen/ewl_fullscreen.c File Reference	809
8.9	bin/tests/image_thumbnail/ewl_image_thumbnail.c File Reference	810
8.10	bin/tests/layer/ewl_layer.c File Reference	811
8.11	bin/tests/modal/ewl_modal.c File Reference	812
8.12	bin/tests/pointer/ewl_pointer.c File Reference	813
8.13	bin/tests/puzzle/ewl_puzzle.c File Reference	814
8.14	bin/tests/text_editor/ewl_text_editor.c File Reference	815
8.15	bin/tests/tooltip/ewl_tooltip.c File Reference	816
8.16	lib/Ewl.h File Reference	817
8.17	lib/ewl_attach.c File Reference	821
8.18	lib/ewl_attach.h File Reference	823
8.19	lib/ewl_base.h File Reference	825
8.20	lib/ewl_border.c File Reference	828
8.21	bin/tests/border/ewl_border.c File Reference	829
8.22	lib/ewl_border.h File Reference	830
8.23	lib/ewl_box.c File Reference	832
8.24	bin/tests/box/ewl_box.c File Reference	834
8.25	lib/ewl_box.h File Reference	835
8.26	lib/ewl_button.c File Reference	837
8.27	bin/tests/button/ewl_button.c File Reference	839
8.28	lib/ewl_button.h File Reference	840
8.29	lib/ewl_calendar.c File Reference	842
8.30	bin/tests/calendar/ewl_calendar.c File Reference	844
8.31	lib/ewl_calendar.h File Reference	845
8.32	lib/ewl_callback.c File Reference	847
8.33	lib/ewl_callback.h File Reference	849
8.34	lib/ewl_cell.c File Reference	851
8.35	lib/ewl_cell.h File Reference	852
8.36	lib/ewl_check.c File Reference	853
8.37	lib/ewl_check.h File Reference	854
8.38	lib/ewl_checkbutton.c File Reference	856
8.39	lib/ewl_checkbutton.h File Reference	858
8.40	lib/ewl_colordialog.c File Reference	860
8.41	bin/tests/colordialog/ewl_colordialog.c File Reference	862

8.42 lib/ewl_colordialog.h File Reference	863
8.43 lib/ewl_colorpicker.c File Reference	865
8.44 bin/tests/colorpicker/ewl_colorpicker.c File Reference	869
8.45 lib/ewl_colorpicker.h File Reference	870
8.46 lib/ewl_combo.c File Reference	872
8.47 bin/tests/combo/ewl_combo.c File Reference	874
8.48 lib/ewl_combo.h File Reference	875
8.49 lib/ewl_config.c File Reference	877
8.50 lib/ewl_config.h File Reference	879
8.51 lib/ewl_container.c File Reference	881
8.52 bin/tests/container/ewl_container.c File Reference	885
8.53 lib/ewl_container.h File Reference	886
8.54 lib/ewl_context_menu.c File Reference	890
8.55 lib/ewl_context_menu.h File Reference	892
8.56 lib/ewl_cursor.c File Reference	894
8.57 lib/ewl_cursor.h File Reference	895
8.58 lib/ewl_datepicker.c File Reference	896
8.59 bin/tests/datepicker/ewl_datepicker.c File Reference	898
8.60 lib/ewl_datepicker.h File Reference	899
8.61 lib/ewl_debug.h File Reference	900
8.62 lib/ewl_dialog.c File Reference	906
8.63 bin/tests/dialog/ewl_dialog.c File Reference	908
8.64 lib/ewl_dialog.h File Reference	909
8.65 lib/ewl_dnd.c File Reference	911
8.66 lib/ewl_dnd.h File Reference	913
8.67 lib/ewl_embed.c File Reference	915
8.68 lib/ewl_embed.h File Reference	920
8.69 lib/ewl_engines.c File Reference	924
8.70 lib/ewl_engines.h File Reference	941
8.71 lib/ewl_entry.c File Reference	964
8.72 bin/tests/entry/ewl_entry.c File Reference	967
8.73 lib/ewl_entry.h File Reference	968
8.74 lib/ewl_enums.h File Reference	971
8.75 lib/ewl_events.c File Reference	977
8.76 lib/ewl_events.h File Reference	978
8.77 lib/ewl_expansion.c File Reference	980

8.78 lib/ewl_expansion.h File Reference	982
8.79 lib/ewl_filedialog.c File Reference	983
8.80 bin/tests/filedialog/ewl_filedialog.c File Reference	986
8.81 lib/ewl_filedialog.h File Reference	987
8.82 lib/ewl_filelist.c File Reference	989
8.83 lib/ewl_filelist.h File Reference	992
8.84 lib/ewl_filelist_column.c File Reference	995
8.85 lib/ewl_filelist_column.h File Reference	997
8.86 lib/ewl_filelist_icon.c File Reference	999
8.87 lib/ewl_filelist_icon.h File Reference	1000
8.88 lib/ewl_filelist_list.c File Reference	1002
8.89 lib/ewl_filelist_list.h File Reference	1004
8.90 lib/ewl_filelist_tree.c File Reference	1006
8.91 lib/ewl_filelist_tree.h File Reference	1008
8.92 lib/ewl_filepicker.c File Reference	1010
8.93 bin/tests/filepicker/ewl_filepicker.c File Reference	1013
8.94 lib/ewl_filepicker.h File Reference	1014
8.95 lib/ewl_float.c File Reference	1016
8.96 bin/tests/float/ewl_float.c File Reference	1018
8.97 lib/ewl_float.h File Reference	1019
8.98 lib/ewl_freebox.c File Reference	1021
8.99 bin/tests/freebox/ewl_freebox.c File Reference	1023
8.100 lib/ewl_freebox.h File Reference	1024
8.101 lib/ewl_grid.c File Reference	1026
8.102 bin/tests/grid/ewl_grid.c File Reference	1029
8.103 lib/ewl_grid.h File Reference	1030
8.104 lib/ewl_highlight.c File Reference	1033
8.105 lib/ewl_highlight.h File Reference	1034
8.106 lib/ewl_histogram.c File Reference	1036
8.107 bin/tests/histogram/ewl_histogram.c File Reference	1038
8.108 lib/ewl_histogram.h File Reference	1039
8.109 lib/ewl_icon.c File Reference	1041
8.110 bin/tests/icon/ewl_icon.c File Reference	1044
8.111 lib/ewl_icon.h File Reference	1045
8.112 lib/ewl_icon_theme.c File Reference	1048
8.113 bin/tests/icon_theme/ewl_icon_theme.c File Reference	1049

8.114lib/ewl_icon_theme.h File Reference	1050
8.115lib/ewl_image.c File Reference	1057
8.116bin/tests/image/ewl_image.c File Reference	1060
8.117lib/ewl_image.h File Reference	1061
8.118lib/ewl_io_manager.c File Reference	1064
8.119bin/tests/io_manager/ewl_io_manager.c File Reference	1068
8.120lib/ewl_io_manager.h File Reference	1069
8.121lib/ewl_label.c File Reference	1073
8.122lib/ewl_label.h File Reference	1074
8.123lib/ewl_list.c File Reference	1076
8.124bin/tests/list/ewl_list.c File Reference	1077
8.125lib/ewl_list.h File Reference	1078
8.126lib/ewl_macros.h File Reference	1079
8.127lib/ewl_media.c File Reference	1082
8.128bin/tests/media/ewl_media.c File Reference	1084
8.129lib/ewl_media.h File Reference	1085
8.130lib/ewl_menu.c File Reference	1087
8.131bin/tests/menu/ewl_menu.c File Reference	1089
8.132lib/ewl_menu.h File Reference	1090
8.133lib/ewl_menu_item.c File Reference	1092
8.134lib/ewl_menu_item.h File Reference	1093
8.135lib/ewl_menubar.c File Reference	1095
8.136bin/tests/menubar/ewl_menubar.c File Reference	1096
8.137lib/ewl_menubar.h File Reference	1097
8.138lib/ewl_misc.c File Reference	1099
8.139lib/ewl_misc.h File Reference	1102
8.140lib/ewl_model.c File Reference	1103
8.141lib/ewl_model.h File Reference	1106
8.142lib/ewl_mvc.c File Reference	1109
8.143bin/tests/mvc/ewl_mvc.c File Reference	1112
8.144lib/ewl_mvc.h File Reference	1113
8.145lib/ewl_notebook.c File Reference	1117
8.146bin/tests/notebook/ewl_notebook.c File Reference	1119
8.147lib/ewl_notebook.h File Reference	1120
8.148lib/ewl_object.c File Reference	1122
8.149lib/ewl_object.h File Reference	1127

8.150lib/ewl_overlay.c File Reference	1133
8.151lib/ewl_overlay.h File Reference	1134
8.152lib/ewl_paned.c File Reference	1135
8.153bin/tests/paned/ewl_paned.c File Reference	1137
8.154lib/ewl_paned.h File Reference	1138
8.155lib/ewl_password.c File Reference	1140
8.156bin/tests/password/ewl_password.c File Reference	1142
8.157lib/ewl_password.h File Reference	1143
8.158lib/ewl_popup.c File Reference	1145
8.159lib/ewl_popup.h File Reference	1147
8.160lib/ewl_private.h File Reference	1149
8.161lib/ewl_progressbar.c File Reference	1154
8.162bin/tests/progressbar/ewl_progressbar.c File Reference	1156
8.163lib/ewl_progressbar.h File Reference	1157
8.164lib/ewl_radiobutton.c File Reference	1159
8.165lib/ewl_radiobutton.h File Reference	1160
8.166lib/ewl_range.c File Reference	1162
8.167lib/ewl_range.h File Reference	1164
8.168lib/ewl_row.c File Reference	1166
8.169lib/ewl_row.h File Reference	1168
8.170lib/ewl_scrollbar.c File Reference	1170
8.171bin/tests/scrollbar/ewl_scrollbar.c File Reference	1172
8.172lib/ewl_scrollbar.h File Reference	1173
8.173lib/ewl_scrollpane.c File Reference	1175
8.174bin/tests/scrollpane/ewl_scrollpane.c File Reference	1177
8.175lib/ewl_scrollpane.h File Reference	1178
8.176lib/ewl_seeker.c File Reference	1180
8.177bin/tests/seeker/ewl_seeker.c File Reference	1182
8.178lib/ewl_seeker.h File Reference	1183
8.179lib/ewl_separator.c File Reference	1185
8.180lib/ewl_separator.h File Reference	1186
8.181lib/ewl_shadow.c File Reference	1188
8.182bin/tests/shadow/ewl_shadow.c File Reference	1189
8.183lib/ewl_shadow.h File Reference	1190
8.184lib/ewl_spacer.c File Reference	1191
8.185lib/ewl_spacer.h File Reference	1192

8.186lib/ewl_spectrum.c File Reference	1193
8.187lib/ewl_spectrum.h File Reference	1195
8.188lib/ewl_spinner.c File Reference	1197
8.189bin/tests/spinner/ewl_spinner.c File Reference	1199
8.190lib/ewl_spinner.h File Reference	1200
8.191lib/ewl_statusbar.c File Reference	1202
8.192bin/tests/statusbar/ewl_statusbar.c File Reference	1204
8.193lib/ewl_statusbar.h File Reference	1205
8.194lib/ewl_stock.c File Reference	1207
8.195lib/ewl_stock.h File Reference	1209
8.196lib/ewl_table.c File Reference	1211
8.197bin/tests/table/ewl_table.c File Reference	1213
8.198lib/ewl_table.h File Reference	1214
8.199lib/ewl_text.c File Reference	1216
8.200bin/tests/text/ewl_text.c File Reference	1225
8.201lib/ewl_text.h File Reference	1226
8.202lib/ewl_theme.c File Reference	1234
8.203bin/tests/theme/ewl_theme.c File Reference	1236
8.204lib/ewl_theme.h File Reference	1237
8.205lib/ewl_toolbar.c File Reference	1238
8.206bin/tests/toolbar/ewl_toolbar.c File Reference	1239
8.207lib/ewl_toolbar.h File Reference	1240
8.208lib/ewl_tree.c File Reference	1242
8.209bin/tests/tree/ewl_tree.c File Reference	1246
8.210lib/ewl_tree.h File Reference	1247
8.211lib/ewl_tree2.c File Reference	1250
8.212bin/tests/tree2/ewl_tree2.c File Reference	1253
8.213lib/ewl_tree2.h File Reference	1255
8.214lib/ewl_tree2_view.c File Reference	1258
8.215lib/ewl_tree2_view.h File Reference	1259
8.216lib/ewl_tree2_view_plain.c File Reference	1261
8.217lib/ewl_tree2_view_plain.h File Reference	1262
8.218lib/ewl_tree2_view_scrolled.c File Reference	1263
8.219lib/ewl_tree2_view_scrolled.h File Reference	1264
8.220lib/ewl_view.c File Reference	1266
8.221lib/ewl_view.h File Reference	1268

8.222lib/ewl_widget.c File Reference	1270
8.223bin/tests/widget/ewl_widget.c File Reference	1277
8.224lib/ewl_widget.h File Reference	1278
8.225lib/ewl_window.c File Reference	1284
8.226lib/ewl_window.h File Reference	1288
8.227/home/ewww/e17/libs/ewl/TODO File Reference	1291
 9 Enlightened Widget Library Page Documentation	 1293
9.1 Layering Scheme	1293
9.2 Documentation Images	1294
9.3 How Themes Work	1295
9.4 TODO and Bugs list	1296
9.5 Widget Theme Keys	1299

Chapter 1

The Enlightened Widget Library

1.1 Introduction

The Enlightened Widget Library (EWL) is a widget toolkit based on the libraries developed for Enlightenment 17. Rendering is performed using Evas, a fast abstracted canvas library that supports multiple backends. The appearance of the widgets are described by Edje files, which are essentially files containing a collection of images and descriptions for laying out those images. The goal of EWL is to abstract the use of these backends and to present an easy to use object model to the end programmer.

Overall, EWL is similar in design and functionality to other common toolkits such as GTK+ and QT. The API's differ, but the overall concepts and ideas are similar. If you are familiar with these other toolkits getting into EWL should be relatively simple.

EWL uses the concept of inheritance for describing its widgets. When a class inherits from another class, the functions that operated on the base class can also operate on the inheriting class. For example, in EWL the class **Ewl_Button** (p. 589) inherits from **Ewl_Box** (p. 586), which inherits from **Ewl_Container** (p. 618). This means you can add widgets to the button, just like you could to the box or any other container by using `ewl_container` functions. Since EWL is written in C, it uses a very simple single inheritance system. The first field of the inheriting struct must be the inherited struct, and note, it's not a pointer to the inherited struct. For example:

```
struct Ewl_Foo
{
    Ewl_Bar bar;
    int baz;
};
```

Creates a new class of object, Foo, which inherits from the Bar class and extends it by adding an integer baz. If the first line of the struct had been `Ewl_Bar *bar`; it would most likely result in some buffer overflows when initializing the widget.

1.2 The Object Model

The basis for all widgets in EWL is **Ewl_Object** (p. 709). `Ewl_Objects` are never allocated outside of another widget, it provides size and position information for the widget as well as info about the widgets padding and insets. There are also fields for indicating object alignment and fill policies.

The next step above **Ewl_Object** (p. 709) is **Ewl_Widget** (p. 785). Again, **Ewl_Widgets** are usually not allocated by themselves, but are part of all the other widgets available in EWL. The **Ewl_Widget** (p. 785) class provides the necessary information about a widget that relates to its appearance, its parent container, event handling, as well as a few miscellaneous tasks common to all widgets. The **Ewl_Widget** (p. 785) structure also contains information on the inheritance of any given widget. This is useful if you need to check the type of a widget in your code.

A necessary class that some widgets inherit from is **Ewl_Container** (p. 618). This is used for holding collections of widgets and specifying their layout. Containers are the building blocks of the widget set. They allow for creating hierarchies of widgets that are bounded within their parent containers. **Ewl_Container** (p. 618) inherits from **Ewl_Widget** (p. 785), so any container can also be treated as a widget, and thus you can put containers within other containers. Examples of inheriting classes are **Ewl_Window** (p. 788) and **Ewl_Box** (p. 586). In the case of **Ewl_Window** (p. 788), widgets inside the window are given any position they request within the insets of the window. For **Ewl_Box** (p. 586), contained widgets are layed out either from top to bottom, or from left to right, depending on the box orientation.

1.3 Callbacks

To do work in a GUI, it is necessary to know when certain actions occur. EWL handles notification of actions using a common technique called callbacks. When the end programmer wants to know when a specific event occurs to a widget, they can add a callback to it using **ewl_callback_append()** (p. 47) or one of the similar functions. One of the arguments to these functions is a pointer to a function. This function will get called when EWL receives the specified event on that widget. You can attach callbacks to any widget, and with containers you can even mark them to intercept events to their child widgets.

One feature of EWL that is different from other toolkits is that it makes extensive use of internal callbacks. In fact, almost all appearance changes for widgets are actually callbacks, and most of the **ewl_widget** calls actually do very little work, but trigger specific callbacks. This feature allows for overriding specific actions on a widget, or for ordering user specified callbacks relative to internal ones.

1.4 Example Application Walk-through

One of the easiest applications to build for EWL is a simple image viewer. The basic image viewer needs a window and an image widget. The following app is a fully functional simple image viewer based on code written by Ben Rockwood of Cuddletech. The first part necessary for creating an EWL application is to include the necessary header **Ewl.h** (p. 817). Following the include statements are global variable declarations.

```
#include <Ewl.h>

Ewl_Widget *main_win;
Ewl_Widget *main_box;
Ewl_Widget *image;
```

Now declarations of function callbacks are made, normally when writing an application these are added after the GUI code is written. The next piece of code is common to most apps, the windows in EWL are not closed unless they are destroyed, so a callback must be attached for the windows delete callback.

```
void
```

```

__destroy_main_window(Ewl_Widget *main_win, void *ev_data, void *user_data)
{
    ewl_widget_destroy(main_win);
    ewl_main_quit();

    return;
}

```

For this application that is the only callback, now we have the main function. This is where EWL is initialized, widgets are created, and the main EWL loop is started. First, declare the main function and check to be sure that an image file was specified. Then, initialize EWL.

```

int main (int argc, char **argv)
{
    if (argc < 2) {
        fprintf(stderr, "Usage: %s <image>\n", argv[0]);
        return 1;
    }
    if (!ewl_init(&argc, argv)) {
        fprintf(stderr, "Unable to initialize EWL.\n");
        return 1;
    }
}

```

Next allocate the window, set its title and attach a callback to catch its delete event. Also, set a minimum size on the window, mark it to be auto-sized, and set visible. Marking it auto-sized will cause the widget to resize to fit the contents.

```

main_win = ewl_window_new();
ewl_window_title_set(EWL_WINDOW(main_win), "EWL Simple Image Viewer");
ewl_callback_append(main_win, EWL_CALLBACK_DELETE_WINDOW,
    __destroy_main_window, NULL);
ewl_object_minimum_size_set(EWL_OBJECT(main_win), 100, 100);
ewl_widget_show(main_win);

```

Next we create a box to hold the image. This isn't really necessary for this app but demonstrates further how to use containers and makes it easier to add more widgets later.

```

main_box = ewl_vbox_new();
ewl_container_child_append(EWL_CONTAINER(main_win), main_box);
ewl_widget_show(main_box);

```

Next, create the image widget. We just attempt to load the image file that was specified on the command line and add it to the box in the window. The second argument is NULL for normal images but can be set to the name of the group to load for an Edje file.

```

image = ewl_image_new();
ewl_image_file_set(EWL_IMAGE(image), argv[1], NULL);
ewl_container_child_append(EWL_CONTAINER(main_box), image);
ewl_widget_show(image);

```

Finally, we call the main function that starts the EWL event processing loop. With that, our app is complete.

```

    ewl_main();

    return 0;
}

```

Now that the application source has been written, it must be compiled. This is fairly simple with EWL, if you name the app `simple_viewer.c` just use the command:

```
gcc -o simple_viewer `ewl-config --cflags --libs` simple_viewer.c
```

1.5 Conclusion

Obviously, creating a simple image viewer doesn't take much effort but it's a good basis for understanding the basics of EWL. Hopefully, readers will extend this app, and possibly create more robust image viewers, and with any luck, other applications to demonstrate EWLs capabilities.

If you have questions, corrections, or improvements, please send them to `enlightenment-devel` at `lists dot sourceforge dot net`.

Chapter 2

Enlightened Widget Library Module Index

2.1 Enlightened Widget Library Modules

Here is a list of all modules:

Ewl_Attach: A mechanism to attach data to widgets.	21
Ewl_Border: A container with a border and label	26
Ewl_Box: The Box Layout Container.	30
Ewl_Button: The Basic Button	35
Ewl_Calendar: The EWL Calendar Widget	41
Ewl_Callback: The Callback Mechanisms	44
Ewl_Cell: The Cell Container, Layout for a Single Widget	51
Ewl_Check: The Simple Check for a Checkbutton	53
Ewl_Checkbutton: An Ewl_Checkbutton with Label	56
Ewl_Colordialog: A Simple Colour Dialog widget	59
Ewl_Colorpicker: The colour picking widget	65
Ewl_Combo: A Simple Combo Box	71
Ewl_Config: Functions for Manipulating Configuration Data	78
Ewl_Container: Widgets Holding Other Widgets	84
Ewl_Context_Menu: A context menu	100
Ewl_Cursor: Container to create custom mouse cursors	104
Ewl_Datepicker: The EWL Datepicker widget	106
Ewl_Dialog: A dialog window	108
Ewl_Dnd: The files containing DND functions	112
Ewl_Embed: A Container for Displaying on an canvas	118
Ewl_Entry: A text entry widget	136
Ewl_Enums: Various Flags and Enumerations used in EWL	144
Ewl_Events: Lower Level Event Handlers	166
Ewl_Expansion: The Expansion node for Tree2	170
Ewl_Filedialog: A Dialog For Picking Files	173
Ewl_Filelist: The base widget for the filelists	180
Ewl_Filelist_Column: An expanding column view	190
Ewl_Filelist_Icon: A icon file view	193
Ewl_Filelist_List: A listview of the files	196
Ewl_Filelist_Tree: A treeview of the files	199
Ewl_Filepicker: Allows selecting files from a list	202

Ewl_Floater: A Floating Container	210
Ewl_Freebox: A Freelayout widget	213
Ewl_Grid The ewl grid widget	218
Ewl_Highlight: A highlight widget	229
Ewl_Histogram: A Simple Histogram widget	232
Ewl_Icon: An icon widget	236
Ewl_Icon_Theme: The icon theme code	245
Ewl_Image: An Image Display Widget	276
Ewl_Label: A fully-themable text label	287
Ewl_List: A list widget	290
Ewl_Macros: Useful Macros Used Internally and Available Externally	292
Ewl_Media: A Multi media widget	295
Ewl_Menu: A Simple Windowed Menu	302
Ewl_Menu_Item: The basic menu item	306
Ewl_Menubar: A menu bar widget	308
Ewl_Misc: Miscellaneous Utility Functions	312
Ewl_Model: A data model	315
Ewl_MVC: The base model-view-controller framework	326
Ewl_Notebook: A notebook widget	336
Ewl_Object: Base object. Provides functionality available to all widgets.	342
Ewl_Overlay: A Container for Displaying on a canvas	372
Ewl_Paned: A paned widget	374
Ewl_Password: A Single Line Text Password Widget	379
Ewl_Popup: A override window for the convenience use	383
Ewl_Progressbar: A progressbar from a value	388
Ewl_Radiobutton: A Radio Button Widget and Grouping System	392
Ewl_Range: A Value Selector from a Range	395
Ewl_Row: The Row widget	401
Ewl_Scrollbar: A Simple Scrollbar Widget	404
Ewl_Scrollpane: A Scrollable Viewing Area	409
Ewl_Seeker: A Value Selector from a Range	415
Ewl_Separator: A Visual Separator Between Widgets	419
Ewl_Shadow: A container with a shadow/border	422
Ewl_Spacer: A Widget to Add Space Between Other Widgets	424
Ewl_Spectrum: The colour spectrum widget	426
Ewl_Spinner: A Numerical Value Entry	432
Ewl_Statusbar: A status bar widget	436
Ewl_Stock: The Stock	442
Ewl_Table: The Table Layout Container.	445
Ewl_Text: A text layout widget	452
Ewl_Theme: Methods for Accessing and Modifying Theme Data	486
Ewl_Toolbar: A toolbar widget	490
Ewl_Tree: A Widget for List or Tree Layout	493
Ewl_Tree2: A Widget for List or Tree Layout	505
Ewl_Tree2_View: A tree view	527
Ewl_Tree2_View_Plain: The plain tree view	529
Ewl_Tree2_View_Scrolled The scrolled tree view	531
Ewl_View: A data view	534
Ewl_Widget: The Parent Widget Class Common to All Widgets	539
Ewl_Window: A Container for Displaying in a New Window	563

Chapter 3

Enlightened Widget Library Data Structure Index

3.1 Enlightened Widget Library Data Structures

Here are the data structures with brief descriptions:

Combo_Test_Data	577
Control	578
Ewl_Attach (Contains information about widget attachments)	579
Ewl_Attach_Dnd (DND data associated with the widget)	580
Ewl_Attach_List (A list of things attached to a widget)	581
Ewl_Attach_Tooltip (Inherits from Ewl_Widget (p. 785) and extends to provide information on a tooltip)	582
Ewl_Border (Inherits from Ewl_Box (p. 586) to allow drawing a border and label decoration around widgets)	584
Ewl_Box (Inherits from an Ewl_Container (p. 618) to provide layout facilities for child widgets placed inside)	586
Ewl_Box_Orientation (Provides information on the orientation of a box)	588
Ewl_Button (A simple Ewl_Widget (p. 785) to provide for a clickable button in the UI)	589
Ewl_Calendar (Inherits from an Ewl_Box (p. 586) to provide a calendar widget)	591
Ewl_Callback (Contains information about a callback)	593
Ewl_Callback_Chain (The callback chain contains the length, mask and information on the list)	594
Ewl_Callback_Custom (Contains information about custom callbacks)	595
Ewl_Cell (Inherits from Ewl_Container (p. 618) for packing widgets inside a cell)	596
Ewl_Check (Inherits from Ewl_Widget (p. 785) and extends to provide a checked state)	597
Ewl_Checkbutton (Inherits from Ewl_Button (p. 589) and expands to provide a stateful check button)	598
Ewl_Color_Set (Contains an RGBA set of colours)	600
Ewl_Colordialog (Inherit from the Ewl_Dialog (p. 626) to create a color dialog widget)	601
Ewl_Colorpicker (Inherits from Ewl_Box (p. 586) to produce a color picker widget)	603
Ewl_Colorpicker_Radiobutton (Inherits from Ewl_Radiobutton (p. 725) and extends to provide colorpicker information)	607

Ewl_Colorpicker_Spinner (Inherits from Ewl_Spinner (p. 746) and extends to provide colorpicker information)	609
Ewl_Combo (Inherits from the Ewl_MVC (p. 705) widget to provide a combo widget)	610
Ewl_Combo_Cell (Inherits from Ewl_Cell (p. 596) and is used internally for the combo box)	612
Ewl_Config (Contains Ewl configuration data)	614
Ewl_Config_Cache (Contains cached Ewl configuration data)	615
Ewl_Configure_Queue	617
Ewl_Container (Inherits from the Ewl_Widget (p. 785) and expands to allow for placing child widgets within the available space. Also adds notifiers for various child events)	618
Ewl_Context_Menu (Inherits from Ewl_Popup (p. 721) and extends to provide a context menu)	620
Ewl_Cursor (Inherits from Ewl_Window (p. 788) to allow packing widgets as cursor objects)	622
Ewl_Datepicker (Inherits from Ewl_Text (p. 757) and extends to provide a date picker)	624
Ewl_Dialog (Inherits from Ewl_Window (p. 788) and extends to provide a dialog widget)	626
Ewl_Dnd_Types (Provides type information about an external DND drag)	628
Ewl_Embed (Inherits from Ewl_Overlay (p. 712) that acts as a top level widget for interacting with the canvas)	629
Ewl_Engine (Contains information on an Ewl engine)	633
Ewl_Engine_Info (Contains the function pointers provided by an engine)	634
Ewl_Entry (Inherits from Ewl_Text (p. 757) and extends to allow editing of the text)	635
Ewl_Entry_Cursor (Inherit from Ewl_Widget (p. 785) and extend for the Entry cursor)	637
Ewl_Event_Action_Response (Stores the response from a user action)	639
Ewl_Event_Dnd_Data_Received (Provides information about dnd drop data)	640
Ewl_Event_Dnd_Data_Request (Provides information about dnd data requests)	641
Ewl_Event_Dnd_Drop (Provides information about dnd drops movement)	642
Ewl_Event_Dnd_Position (Provides information about dnd drops movement)	643
Ewl_Event_Key (Provides the basic information about key events)	644
Ewl_Event_Key_Down (Provides clients with necessary information about the key press event)	645
Ewl_Event_Key_Up (Provides clients with necessary information about the key release event)	646
Ewl_Event_Mouse (Provides base mouse information)	647
Ewl_Event_Mouse_Down (Provides information about the mouse down event)	648
Ewl_Event_Mouse_In (Provides information about the mouse entering)	649
Ewl_Event_Mouse_Move (Provides information about mouse movement)	650
Ewl_Event_Mouse_Out (Provides information about the mouse leaving)	651
Ewl_Event_Mouse_Up (Provides information about the mouse up event)	652
Ewl_Event_Mouse_Wheel (Provides information about the mouse wheel scrolling)	653
Ewl_Event_Window_Expose (Describes the canvas region that received an expose notification)	654
Ewl_Expansion (Inherits from Ewl_Check (p. 597) and extends it to provide a expanded state)	655
Ewl_Filedialog (Inherits from Ewl_Dialog (p. 626) and extends to create a filedialog)	656
Ewl_Filelist (Inherits from Ewl_Box (p. 586) and extends to display a file list)	658
Ewl_Filelist_Column (Inherits from Ewl_Filelist (p. 658) and extends to provide a column view)	661
Ewl_Filelist_Column_Data (Stores information on the filelist column data)	663
Ewl_Filelist_Icon (Inherits from Ewl_Filelist (p. 658) and extends for an icon view)	664

Ewl_Filelist_List (Inherits from Ewl_Filelist (p. 658) and extends to create a list view)	665
Ewl_Filelist_Tree (Inherits from Ewl_Filelist (p. 658) and extends to provide a tree view)	666
Ewl_Filelist_Tree_Data (Contains information on a filelist tree data)	667
Ewl_Filepicker (Inherits from Ewl_Box (p. 586) and extends to provide a filepicker)	668
Ewl_Filepicker_Filter (Contains information on a filepicker filter)	670
Ewl_Floater (Inherits from Ewl_Box (p. 586) for layout purposes. Provides a means to layer above other widgets and to follow the movement of another widget) .	671
Ewl_Freebox (Inherits from Ewl_Container (p. 618) and extends to allow for different layout algorithms)	673
Ewl_Grid (Inherit from Ewl_Container (p. 618) and extend to provide a grid layout widget)	675
Ewl_Grid_Child (Contains information about a grid child)	677
Ewl_Grid_Info (Contains information about a row or column of Ewl_Grid (p. 675))	678
Ewl_Highlight (Inherits from Ewl_Floater (p. 671) and extends to provide a highlight widget)	680
Ewl_Histogram (Inherit from the Ewl_Image (p. 686) and extends to create a histogram widget)	681
Ewl_Icon (Inherits from Ewl_Stock (p. 750) and extends to provide for an icon layout)	683
Ewl_Image (Inherits from Ewl_Widget (p. 785) and extends to provide an image widget)	686
Ewl_Image_Thumbnail (Inherits from Ewl_Image (p. 686) and extends to provide reference to original image)	689
Ewl_IO_Manager_Plugin (Provides the base structure for IO Manager plugins) .	690
Ewl_Label (Inherits from Ewl_Widget (p. 785) class to provide a simple label) . .	692
Ewl_List (Inherits from Ewl_MVC (p. 705) and extends to provide a list widget) .	693
Ewl_Media (Inherits from the Ewl_Widget (p. 785) class and extends it to provide for multi-line media layout, obstacle wrapping, and a variety of formatting) . .	694
Ewl_Menu (Inherits from the Ewl_Menu_Base and extends to provide a simple menu)	696
Ewl_Menu_Info (Provides a simple structure to contain menu information)	698
Ewl_Menu_Item (Inherits from Ewl_Button (p. 589) and provides a basic menu item widget)	699
Ewl_Menubar (Inherits from the Ewl_Box (p. 586) class and extends it to provide for a menubar)	701
Ewl_Menubar_Info (Provides a simple menu structure for the menubar)	702
Ewl_Model (Holds the callbacks needed to define a model)	703
Ewl_MVC (Inherits from Ewl_Box (p. 586) and extends to provide mvc functionality)	705
Ewl_Notebook (Inherits from Ewl_Box (p. 586) and extends to provide for a notebook style layout)	707
Ewl_Object (Provides facilities for sizing, position, alignment and fill policy)	709
Ewl_Overlay (The class inheriting from Ewl_Container (p. 618) that acts as a top level widget for interacting with the evas)	712
Ewl_Pair (Contains a key and a value pair)	713
Ewl_Pair_List (Contains a list of key value pairs)	714
Ewl_Paned (Inherits from Ewl_Container (p. 618) and extends to provide the paned widget)	715
Ewl_Paned_Grabber (Inherits from Ewl_Separator (p. 740) and extends to facilitate the paned grabber)	717
Ewl_Paned_Layout (Contains information on a paned layout)	718
Ewl_Password (Inherits from the Ewl_Entry (p. 635) and extends it to obscure the text)	719
Ewl_Popup (Inherits from Ewl_Window (p. 788) and extends to provide a popup widget)	721

Ewl_Progressbar (Inherits from Ewl_Range (p. 727) and extends to provide a progressbar widget)	723
Ewl_Radiobutton (Inherits from Ewl_Checkbutton (p. 598) and extends it to provide grouping buttons to limit to a single selection in a group at a given time)	725
Ewl_Range (Inherits from Ewl_Container (p. 618) and extends to provide a basic range widget)	727
Ewl_Row (Inherits from Ewl_Container (p. 618) to create a tree row that lays out cells in a left to right fashion)	729
Ewl_Scrollbar (Inherits from Ewl_Box (p. 586) and extends to provide a scrollbar widget)	730
Ewl_Scrollpane (Inherits from Ewl_Container (p. 618) and extends it to enclose a single widget in vertical and horizontal scrollbars for panning around displaying different regions of the enclosed widget)	732
Ewl_Seeker (Inherits from Ewl_Range (p. 727) and extends to provide a seeker widget)	734
Ewl_Selection (Structure to store information on selections)	736
Ewl_Selection_Idx (Structure to store information on a single index selection)	737
Ewl_Selection_Range (Structure to store a range of selected cells)	738
Ewl_Separator (Inherits from Ewl_Widget (p. 785) to allow drawing a horizontal or vertical line between widgets, depending on the orientation the Ewl_Separator (p. 740))	740
Ewl_Shadow (Inherits from Ewl_Box (p. 586) to allow drawing a border and shadow decoration around widgets)	741
Ewl_Spacer (Inherits from Ewl_Widget (p. 785) and provides as a spacer between widgets)	742
Ewl_Spectrum (Inherits from Ewl_Overlay (p. 712) and extends to provide a colour spectrum)	743
Ewl_Spinner (Inherits from Ewl_Range (p. 727) and extends to provide a spinner widget)	746
Ewl_Statusbar (Inherits from the Ewl_Box (p. 586) class and extends it to provide for a simple status bar)	748
Ewl_Stock (Inherits from Ewl_Box (p. 586) and provides the base for the other stock icons)	750
Ewl_Stock_Funcs (The function callbacks that an inheriting widget needs to implement)	752
Ewl_Table (Inherits from Ewl_Container (p. 618) and extends to provide a table layout)	753
Ewl_Test (Holds information on an Ewl Test set)	755
Ewl_Text (Inherits from the Ewl_Container (p. 618) class and extends it to provide text layout and formatting)	757
Ewl_Text_Context (Stores context information for the different formatting nodes)	761
Ewl_Text_Fmt (Text formatting information)	764
Ewl_Text_Trigger (Inherits from Ewl_Widget (p. 785) and extends to provide a trigger for the text widget)	765
Ewl_Text_Trigger_Area (Inherits from Ewl_Widget (p. 785) and extends to provide a trigger area)	767
Ewl_Toolbar (Inherits from Ewl_Menuubar (p. 701) and extends to provide a toolbar layout)	768
Ewl_Tree (Inherits from Ewl_Container (p. 618) and extends to provide a tree layout)	769
Ewl_Tree2 (Inherits from Ewl_MVC (p. 705) and extends to provide a tree widget)	771
Ewl_Tree2_Node (Inherits from Ewl_Container (p. 618) and extends to hold information on a row of the tree)	774

Ewl_Tree2_View (Inherits from Ewl_Box (p. 586) and provides the base information for a tree view)	776
Ewl_Tree2_View_Plain (Inherits from Ewl_Tree2_View (p. 776) and provides a plain tree layout)	778
Ewl_Tree2_View_Scrolled (Inherits from Ewl_Box (p. 586) and provides a scrolled tree layout)	779
Ewl_Tree_Node (Inherits from Ewl_Container (p. 618) and extends to hold information on a row of the tree)	781
Ewl_Unit_Test (Contains the information needed to run a unit test)	783
Ewl_View (The view function pointers)	784
Ewl_Widget (Inherits from Ewl_Object (p. 709) and extends to provide appearance, parent, and callback capabilities)	785
Ewl_Window (Inherits from Ewl_Embed (p. 629) class to create it's own window and canvas for drawing, sizing and positioning)	788
Freebox_Test	790
List_Test_Data	791
List_Test_Row_Data	792
Tree2_Test_Data	793
Tree2_Test_Row_Data	794

Chapter 4

Enlightened Widget Library File Index

4.1 Enlightened Widget Library File List

Here is a list of all files with brief descriptions:

/home/ewww/e17/libs/ewl/ TODO	1291
bin/ ewl_embed_test.c	795
bin/ ewl_simple_test.c	797
bin/ Ewl_Test.h	798
bin/ ewl_test_private.h	801
bin/ main.c	804
bin/tests/border/ ewl_border.c	829
bin/tests/box/ ewl_box.c	834
bin/tests/button/ ewl_button.c	839
bin/tests/calendar/ ewl_calendar.c	844
bin/tests/colordialog/ ewl_colordialog.c	862
bin/tests/colorpicker/ ewl_colorpicker.c	869
bin/tests/combo/ ewl_combo.c	874
bin/tests/container/ ewl_container.c	885
bin/tests/datepicker/ ewl_datepicker.c	898
bin/tests/dialog/ ewl_dialog.c	908
bin/tests/dnd_snoop/ ewl_dnd_snoop.c	806
bin/tests/engine/ ewl_engine.c	808
bin/tests/entry/ ewl_entry.c	967
bin/tests/filedialog/ ewl_filedialog.c	986
bin/tests/filepicker/ ewl_filepicker.c	1013
bin/tests/floater/ ewl_floater.c	1018
bin/tests/freebox/ ewl_freebox.c	1023
bin/tests/fullscreen/ ewl_fullscreen.c	809
bin/tests/grid/ ewl_grid.c	1029
bin/tests/histogram/ ewl_histogram.c	1038
bin/tests/icon/ ewl_icon.c	1044
bin/tests/icon_theme/ ewl_icon_theme.c	1049
bin/tests/image/ ewl_image.c	1060
bin/tests/image_thumbnail/ ewl_image_thumbnail.c	810
bin/tests/io_manager/ ewl_io_manager.c	1068

bin/tests/layer/ ewl_layer.c	811
bin/tests/list/ ewl_list.c	1077
bin/tests/media/ ewl_media.c	1084
bin/tests/menu/ ewl_menu.c	1089
bin/tests/menubar/ ewl_menubar.c	1096
bin/tests/modal/ ewl_modal.c	812
bin/tests/mvc/ ewl_mvc.c	1112
bin/tests/notebook/ ewl_notebook.c	1119
bin/tests/paned/ ewl_paned.c	1137
bin/tests/password/ ewl_password.c	1142
bin/tests/pointer/ ewl_pointer.c	813
bin/tests/progressbar/ ewl_progressbar.c	1156
bin/tests/puzzle/ ewl_puzzle.c	814
bin/tests/scrollbar/ ewl_scrollbar.c	1172
bin/tests/scrollpane/ ewl_scrollpane.c	1177
bin/tests/seeker/ ewl_seeker.c	1182
bin/tests/shadow/ ewl_shadow.c	1189
bin/tests/spinner/ ewl_spinner.c	1199
bin/tests/statusbar/ ewl_statusbar.c	1204
bin/tests/table/ ewl_table.c	1213
bin/tests/text/ ewl_text.c	1225
bin/tests/text_editor/ ewl_text_editor.c	815
bin/tests/theme/ ewl_theme.c	1236
bin/tests/toolbar/ ewl_toolbar.c	1239
bin/tests/tooltip/ ewl_tooltip.c	816
bin/tests/tree/ ewl_tree.c	1246
bin/tests/tree2/ ewl_tree2.c	1253
bin/tests/widget/ ewl_widget.c	1277
lib/ Ewl.h (The file that should be included by any project using EWL. Provides all the necessary headers and includes to work with EWL. It is discouraged to include each header file individually)	817
lib/ ewl_attach.c	821
lib/ ewl_attach.h	823
lib/ ewl_base.h	825
lib/ ewl_border.c	828
lib/ ewl_border.h	830
lib/ ewl_box.c	832
lib/ ewl_box.h	835
lib/ ewl_button.c	837
lib/ ewl_button.h	840
lib/ ewl_calendar.c	842
lib/ ewl_calendar.h	845
lib/ ewl_callback.c	847
lib/ ewl_callback.h	849
lib/ ewl_cell.c	851
lib/ ewl_cell.h	852
lib/ ewl_check.c	853
lib/ ewl_check.h	854
lib/ ewl_checkbutton.c	856
lib/ ewl_checkbutton.h	858
lib/ ewl_colordialog.c	860
lib/ ewl_colordialog.h	863
lib/ ewl_colorpicker.c	865
lib/ ewl_colorpicker.h	870

lib/ewl_combo.c	872
lib/ewl_combo.h	875
lib/ewl_config.c	877
lib/ewl_config.h	879
lib/ewl_container.c	881
lib/ewl_container.h	886
lib/ewl_context_menu.c	890
lib/ewl_context_menu.h	892
lib/ewl_cursor.c	894
lib/ewl_cursor.h	895
lib/ewl_datepicker.c	896
lib/ewl_datepicker.h	899
lib/ewl_debug.h	900
lib/ewl_dialog.c	906
lib/ewl_dialog.h	909
lib/ewl_dnd.c	911
lib/ewl_dnd.h	913
lib/ewl_embed.c	915
lib/ewl_embed.h	920
lib/ewl_engines.c	924
lib/ewl_engines.h	941
lib/ewl_entry.c	964
lib/ewl_entry.h	968
lib/ewl_enums.h	971
lib/ewl_events.c	977
lib/ewl_events.h	978
lib/ewl_expansion.c	980
lib/ewl_expansion.h	982
lib/ewl_filedialog.c	983
lib/ewl_filedialog.h	987
lib/ewl_filelist.c	989
lib/ewl_filelist.h	992
lib/ewl_filelist_column.c	995
lib/ewl_filelist_column.h	997
lib/ewl_filelist_icon.c	999
lib/ewl_filelist_icon.h	1000
lib/ewl_filelist_list.c	1002
lib/ewl_filelist_list.h	1004
lib/ewl_filelist_tree.c	1006
lib/ewl_filelist_tree.h	1008
lib/ewl_filepicker.c	1010
lib/ewl_filepicker.h	1014
lib/ewl_float.c	1016
lib/ewl_float.h	1019
lib/ewl_freebox.c	1021
lib/ewl_freebox.h	1024
lib/ewl_grid.c	1026
lib/ewl_grid.h	1030
lib/ewl_highlight.c	1033
lib/ewl_highlight.h	1034
lib/ewl_histogram.c	1036
lib/ewl_histogram.h	1039
lib/ewl_icon.c	1041
lib/ewl_icon.h	1045

lib/ewl_icon_theme.c	1048
lib/ewl_icon_theme.h	1050
lib/ewl_image.c	1057
lib/ewl_image.h	1061
lib/ewl_io_manager.c	1064
lib/ewl_io_manager.h	1069
lib/ewl_label.c	1073
lib/ewl_label.h	1074
lib/ewl_list.c	1076
lib/ewl_list.h	1078
lib/ewl_macros.h	1079
lib/ewl_media.c	1082
lib/ewl_media.h	1085
lib/ewl_menu.c	1087
lib/ewl_menu.h	1090
lib/ewl_menu_item.c	1092
lib/ewl_menu_item.h	1093
lib/ewl_menubar.c	1095
lib/ewl_menubar.h	1097
lib/ewl_misc.c	1099
lib/ewl_misc.h	1102
lib/ewl_model.c	1103
lib/ewl_model.h	1106
lib/ewl_mvc.c	1109
lib/ewl_mvc.h	1113
lib/ewl_notebook.c	1117
lib/ewl_notebook.h	1120
lib/ewl_object.c	1122
lib/ewl_object.h	1127
lib/ewl_overlay.c	1133
lib/ewl_overlay.h	1134
lib/ewl_paned.c	1135
lib/ewl_paned.h	1138
lib/ewl_password.c	1140
lib/ewl_password.h	1143
lib/ewl_popup.c	1145
lib/ewl_popup.h	1147
lib/ewl_private.h	1149
lib/ewl_progressbar.c	1154
lib/ewl_progressbar.h	1157
lib/ewl_radiobutton.c	1159
lib/ewl_radiobutton.h	1160
lib/ewl_range.c	1162
lib/ewl_range.h	1164
lib/ewl_row.c	1166
lib/ewl_row.h	1168
lib/ewl_scrollbar.c	1170
lib/ewl_scrollbar.h	1173
lib/ewl_scrollpane.c	1175
lib/ewl_scrollpane.h	1178
lib/ewl_seeker.c	1180
lib/ewl_seeker.h	1183
lib/ewl_separator.c	1185
lib/ewl_separator.h	1186

lib/ewl_shadow.c	1188
lib/ewl_shadow.h	1190
lib/ewl_spacer.c	1191
lib/ewl_spacer.h	1192
lib/ewl_spectrum.c	1193
lib/ewl_spectrum.h	1195
lib/ewl_spinner.c	1197
lib/ewl_spinner.h	1200
lib/ewl_statusbar.c	1202
lib/ewl_statusbar.h	1205
lib/ewl_stock.c	1207
lib/ewl_stock.h	1209
lib/ewl_table.c	1211
lib/ewl_table.h	1214
lib/ewl_text.c	1216
lib/ewl_text.h	1226
lib/ewl_theme.c	1234
lib/ewl_theme.h	1237
lib/ewl_toolbar.c	1238
lib/ewl_toolbar.h	1240
lib/ewl_tree.c	1242
lib/ewl_tree.h	1247
lib/ewl_tree2.c	1250
lib/ewl_tree2.h	1255
lib/ewl_tree2_view.c	1258
lib/ewl_tree2_view.h	1259
lib/ewl_tree2_view_plain.c	1261
lib/ewl_tree2_view_plain.h	1262
lib/ewl_tree2_view_scrolled.c	1263
lib/ewl_tree2_view_scrolled.h	1264
lib/ewl_view.c	1266
lib/ewl_view.h	1268
lib/ewl_widget.c	1270
lib/ewl_widget.h	1278
lib/ewl_window.c	1284
lib/ewl_window.h	1288

Chapter 5

Enlightened Widget Library Page Index

5.1 Enlightened Widget Library Related Pages

Here is a list of all related documentation pages:

Layering Scheme	1293
Documentation Images	1294
How Themes Work	1295
TODO and Bugs list	1296
Widget Theme Keys	1299

Chapter 6

Enlightened Widget Library Module Documentation

6.1 Ewl_Attach: A mechanism to attach data to widgets.

6.1.1 Detailed Description

Defines methods for attaching data to widgets.

Data Structures

- struct **Ewl_Attach**
Contains information about widget attachments.
- struct **Ewl_Attach_Dnd**
DND data associated with the widget.

Defines

- #define **EWL_ATTACH**(attach) ((**Ewl_Attach** *) attach)
- #define **ewl_attach_color_get**(w) ewl_attach_get(EWL_WIDGET(w), EWL_ATTACH_TYPE_COLOR)
- #define **ewl_attach_color_set**(w, data) ewl_attach_other_set(EWL_WIDGET(w), EWL_ATTACH_TYPE_COLOR, data)
- #define **EWL_ATTACH_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w)), EWL_ATTACH_TYPE)
- #define **ewl_attach_mouse_argb_cursor_set**(w, cursor)
- #define **ewl_attach_mouse_cursor_set**(w, data)
- #define **ewl_attach_name_get**(w) ewl_attach_get(EWL_WIDGET(w), EWL_ATTACH_TYPE_NAME)
- #define **ewl_attach_name_set**(w, data) ewl_attach_text_set(EWL_WIDGET(w), EWL_ATTACH_TYPE_NAME, data)
- #define **ewl_attach_tooltip_text_set**(w, data) ewl_attach_text_set(EWL_WIDGET(w), EWL_ATTACH_TYPE_TOOLTIP, data)

- `#define EWL_ATTACH_TOOLTIP_TYPE "tooltip"`
- `#define ewl_attach_tooltip_widget_set(w, data) ewl_attach_widget_set(EWL_WIDGET(w), EWL_ATTACH_TYPE_TOOLTIP, data)`
- `#define ewl_attach_widget_association_get(w) ewl_attach_get(EWL_WIDGET(w), EWL_ATTACH_TYPE_WIDGET_ASSOCIATION)`
- `#define ewl_attach_widget_association_set(w, data)`

Typedefs

- `typedef Ewl_Attach Ewl_Attach`
- `typedef Ewl_Attach_Dnd Ewl_Attach_Dnd`

Functions

- `void * ewl_attach_get (Ewl_Widget *w, Ewl_Attach_Type t)`
Get the attachment of type `t` from the widget `w`.
- `void ewl_attach_list_del (Ewl_Attach_List *list, Ewl_Attach_Type type)`
Deletes the given type `type` from the list `list`.
- `void ewl_attach_other_set (Ewl_Widget *w, Ewl_Attach_Type t, void *data)`
Attaches the data `data` to the widget `w` with the attache type of `t`.
- `void ewl_attach_text_set (Ewl_Widget *w, Ewl_Attach_Type t, const char *data)`
Attaches the text `data` to the widget `w`.
- `void ewl_attach_widget_set (Ewl_Widget *w, Ewl_Attach_Type t, Ewl_Widget *data)`
Attaches a widget `data` to the widget `w`.

6.1.2 Define Documentation

6.1.2.1 `#define EWL_ATTACH(attach) ((Ewl_Attach *) attach)`

Typecast a pointer to an `Ewl_Attach` (p. 579) pointer

6.1.2.2 `#define ewl_attach_color_get(w) ewl_attach_get(EWL_WIDGET(w), EWL_ATTACH_TYPE_COLOR)`

Convenience method to get a colour attachment back

6.1.2.3 `#define ewl_attach_color_set(w, data) ewl_attach_other_set(EWL_WIDGET(w), EWL_ATTACH_TYPE_COLOR, data)`

Convenience method to set a colour widget attachment

6.1.2.4 `#define EWL_ATTACH_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_ATTACH_TYPE)`

Returns TRUE if the widget is an **Ewl_Attach** (p. 579) widget, FALSE otherwise

6.1.2.5 `#define ewl_attach_mouse_argb_cursor_set(w, cursor)`

Value:

```
ewl_attach_other_set(EWL_WIDGET(w), \
                     EWL_ATTACH_TYPE_MOUSE_ARGB_CURSOR, cursor)
```

Convenience method to attach data to a notebook item

6.1.2.6 `#define ewl_attach_mouse_cursor_set(w, data)`

Value:

```
ewl_attach_other_set(EWL_WIDGET(w), EWL_ATTACH_TYPE_MOUSE_CURSOR, \
                     (int *)data)
```

Convenience method to set a cursor widget attachment

6.1.2.7 `#define ewl_attach_name_get(w) ewl_attach_get(EWL_WIDGET(w), EWL_ATTACH_TYPE_NAME)`

Convenience method to get a widgets name attachment

6.1.2.8 `#define ewl_attach_name_set(w, data) ewl_attach_text_set(EWL_WIDGET(w), EWL_ATTACH_TYPE_NAME, data)`

Convenience method to set a widget name attachment

6.1.2.9 `#define ewl_attach_tooltip_text_set(w, data) ewl_attach_text_set(EWL_WIDGET(w), EWL_ATTACH_TYPE_TOOLTIP, data)`

Convenience method to set a tooltip text attachment

6.1.2.10 `#define EWL_ATTACH_TOOLTIP_TYPE "tooltip"`

The type name for the **Ewl_Attach** (p. 579) tooltip

6.1.2.11 `#define ewl_attach_tooltip_widget_set(w, data) ewl_attach_widget_set(EWL_WIDGET(w), EWL_ATTACH_TYPE_TOOLTIP, data)`

Convenience method to set a tooltip widget attachment

6.1.2.12 `#define ewl_attach_widget_association_get(w) ewl_attach_get(EWL_WIDGET(w), EWL_ATTACH_TYPE_WIDGET_ASSOCIATION)`

Convenience method to get the data attached to a notebook item

6.1.2.13 `#define ewl_attach_widget_association_set(w, data)`

Value:

```
ewl_attach_other_set(EWL_WIDGET(w), \
                    EWL_ATTACH_TYPE_WIDGET_ASSOCIATION, data)
```

Convenience method to attach data to a notebook item

6.1.3 Typedef Documentation

6.1.3.1 `typedef struct Ewl_Attach Ewl_Attach`

`Ewl_Attach` (p. 579) provides a way to attach data to widgets

6.1.3.2 `typedef struct Ewl_Attach_Dnd Ewl_Attach_Dnd`

`Ewl_Attach_Dnd` (p. 580) provides a way to attach DND data to widgets

6.1.4 Function Documentation

6.1.4.1 `void* ewl_attach_get (Ewl_Widget * w, Ewl_Attach_Type t)`

Get the attachment of type `t` from the widget `w`.

Parameters:

- `w`,: The widget to get the attachment from
- `t`,: The type of attachment to get

Returns:

Returns the data for the given attachment type

6.1.4.2 `void ewl_attach_list_del (Ewl_Attach_List * list, Ewl_Attach_Type type)`

Deletes the given type `type` from the list `list`.

Parameters:

- `list`,: The `Ewl_Attach_List` (p. 581) to delete
- `type`,: The `Ewl_Attach_Type` to delete

Returns:

Returns no value

6.1.4.3 void ewl_attach_other_set (Ewl_Widget * *w*, Ewl_Attach_Type *t*, void * *data*)

Attaches the data *data* to the widget *w* with the attach type of *t*.

Parameters:

w,: The widget to attach the data too

t,: The type of attachment

data,: The data to attach

Returns:

Returns no value

6.1.4.4 void ewl_attach_text_set (Ewl_Widget * *w*, Ewl_Attach_Type *t*, const char * *data*)

Attaches the text *data* to the widget *w*.

Parameters:

w,: The widget to attach the text too

t,: The type of the attachment

data,: The text to set as the attachment

Returns:

Returns no value

6.1.4.5 void ewl_attach_widget_set (Ewl_Widget * *w*, Ewl_Attach_Type *t*, Ewl_Widget * *data*)

Attaches a widget *data* to the widget *w*.

Parameters:

w,: The widget to attach the widget too

t,: The type of data being attached

data,: The widget to attach

Returns:

Returns no value

6.2 Ewl_Border: A container with a border and label

6.2.1 Detailed Description

Defines the **Ewl_Border** (p. 584) class used for adding a border decoration around a group of widgets.

Remarks:

Inherits from **Ewl_Box** (p. 586).

Data Structures

- struct **Ewl_Border**

*Inherits from **Ewl_Box** (p. 586) to allow drawing a border and label decoration around widgets.*

Defines

- `#define EWL_BORDER(border) ((Ewl_Border *) border)`
- `#define EWL_BORDER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_BORDER_TYPE))`
- `#define EWL_BORDER_TYPE "border"`

Typedefs

- `typedef Ewl_Border Ewl_Border`

Functions

- `int ewl_border_init (Ewl_Border *b)`
Initialize a border container to default values.
- `unsigned int ewl_border_label_alignment_get (Ewl_Border *b)`
Retruns the alignment setting of the label for this border container.
- `void ewl_border_label_alignment_set (Ewl_Border *b, unsigned int align)`
alters the alignment setting of the label on the border
- `const char * ewl_border_label_get (Ewl_Border *b)`
Get the label from a border widget.
- `Ewl_Position ewl_border_label_position_get (Ewl_Border *b)`
Gets the label position of the border.
- `void ewl_border_label_position_set (Ewl_Border *b, Ewl_Position pos)`
Sets the position of the label in the border container.
- `void ewl_border_label_set (Ewl_Border *b, const char *t)`

Set the label for the border.

- **Ewl_Widget * ewl_border_new** (void)
Allocate and initialize a new border container.

6.2.2 Define Documentation

6.2.2.1 #define EWL_BORDER(border) ((Ewl_Border *) border)

Typecast a pointer to an **Ewl_Separator** (p. 740) pointer.

6.2.2.2 #define EWL_BORDER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_BORDER_TYPE))

Returns TRUE if the widget is an **Ewl_Border** (p. 584) widget, FALSE otherwise

6.2.2.3 #define EWL_BORDER_TYPE "border"

The type name for the **Ewl_Border** (p. 584) widget

6.2.3 Typedef Documentation

6.2.3.1 typedef struct Ewl_Border Ewl_Border

Ewl_Border (p. 584) provides a container that has a labeled border.

6.2.4 Function Documentation

6.2.4.1 int ewl_border_init (Ewl_Border * b)

Initialize a border container to default values.

Parameters:

b,: the border container to initialize

Returns:

Returns TRUE on success, FALSE on failure.

6.2.4.2 unsigned int ewl_border_label_alignment_get (Ewl_Border * b)

Retruns the alignment setting of the label for this border container.

Parameters:

b,: The **Ewl_Border** (p. 584) to get the alignment from

Returns:

Returns the alignment of the label for the border.

6.2.4.3 void ewl_border_label_alignment_set (Ewl_Border * *b*, unsigned int *align*)

alters the alignment setting of the label on the border

Parameters:

b,: The **Ewl_Border** (p. 584) to set the alignment on

align,: The alignment to set on the label

Returns:

Retruns no value

6.2.4.4 const char* ewl_border_label_get (Ewl_Border * *b*)

Get the label from a border widget.

Parameters:

b,: the border to retrieve the label text

Returns:

Returns the border label text on success, NULL on failure.

6.2.4.5 Ewl_Position ewl_border_label_position_get (Ewl_Border * *b*)

Gets the label position of the border.

Parameters:

b,: The **Ewl_Border** (p. 584) to get the label position from

Returns:

Returns the Ewl_Position setting of the label on this border

6.2.4.6 void ewl_border_label_position_set (Ewl_Border * *b*, Ewl_Position *pos*)

Sets the position of the label in the border container.

Parameters:

b,: The **Ewl_Border** (p. 584) to set the label position on

pos,: The Ewl_Position to set on for the label.

Returns:

Returns no value.

6.2.4.7 void ewl_border_label_set (Ewl_Border * *b*, const char * *t*)

Set the label for the border.

Parameters:

b,: the border widget to change the text

t,: the text to set for the border label

Returns:

Returns no value.

Change the text of the border label to the string *t*.

6.2.4.8 Ewl_Widget* ewl_border_new (void)

Allocate and initialize a new border container.

Returns:

Returns a new border container on success, NULL on failure.

6.3 Ewl_Box: The Box Layout Container.

6.3.1 Detailed Description

Defines the **Ewl_Box** (p. 586) class used for laying out Ewl_Widget's in a horizontal or vertical line.

Remarks:

Inherits from **Ewl_Container** (p. 618).

6.3.2 Tutorial

The **Ewl_Box** (p. 586) provides a simple linear layout container. There are a few options to control the layout characteristics, but much of the layout is dependent on the child widget settings. The first option encountered is the orientation, which is either horizontal or vertical. Both orientations are implemented within the same box widget, and is set with `ewl_box_orientation_set`, but shortcuts exist in the form of `ewl_hbox_new` and `ewl_vbox_new`. The default orientation is horizontal.

```
Ewl_Widget *box;
box = ewl_hbox_new();
ewl_widget_show(box);
```

After creating the box, children are added with the standard container calls common to most containers. For example, a label and entry would be added as follows.

```
ewl_container_child_append(EWL_CONTAINER(box), label);
ewl_container_child_append(EWL_CONTAINER(box), entry);
```

This would result in a label followed by the entry laid out horizontally.

Data Structures

- struct **Ewl_Box**

*Inherits from an **Ewl_Container** (p. 618) to provide layout facilities for child widgets placed inside.*

Defines

- `#define EWL_BOX(box) ((Ewl_Box *) box)`
- `#define EWL_BOX_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_BOX_TYPE))`
- `#define EWL_BOX_TYPE "box"`

Typedefs

- `typedef Ewl_Box Ewl_Box`

Functions

- void **ewl_box_cb_child_hide** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_box_cb_child_homogeneous_show** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_box_cb_child_resize** (**Ewl_Container** *c, **Ewl_Widget** *w, int size, **Ewl_Orientation** o)
- void **ewl_box_cb_child_show** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_box_cb_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_box_cb_configure_homogeneous** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_box_homogeneous_set** (**Ewl_Box** *b, unsigned int h)
Change the homogeneous layout of the box.
- int **ewl_box_init** (**Ewl_Box** *box)
Initialize the box to starting values.
- **Ewl_Widget** * **ewl_box_new** (void)
Allocate and initialize a new box with given orientation.
- **Ewl_Orientation** **ewl_box_orientation_get** (**Ewl_Box** *b)
Retrieves the orientation of the box.
- void **ewl_box_orientation_set** (**Ewl_Box** *b, **Ewl_Orientation** o)
Change the specified box's orientation.
- void **ewl_box_spacing_set** (**Ewl_Box** *b, int spacing)
Changes the spacing between the objects in the box.
- **Ewl_Widget** * **ewl_hbox_new** (void)
Allocate and initialize a new box with horizontal orientation.
- **Ewl_Widget** * **ewl_vbox_new** (void)
Allocate and initialize a new box with vertical orientation.

6.3.3 Define Documentation

6.3.3.1 #define EWL_BOX(box) ((Ewl_Box *) box)

Typecast a pointer to an **Ewl_Box** (p. 586) pointer.

6.3.3.2 #define EWL_BOX_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_BOX_TYPE))

Returns TRUE if the widget is an **Ewl_Box** (p. 586), FALSE otherwise

6.3.3.3 #define EWL_BOX_TYPE "box"

The type name for the **Ewl_Box** (p. 586) widget

6.3.4 Typedef Documentation

6.3.4.1 typedef struct Ewl_Box Ewl_Box

The box widget is an **Ewl_Container** (p. 618) and lays out Ewl_Widget's either horizontally or vertically.

6.3.5 Function Documentation

6.3.5.1 void ewl_box_cb_child_hide (Ewl_Container * *c*, Ewl_Widget * *w*)

6.3.5.2 void ewl_box_cb_child_homogeneous_show (Ewl_Container * *c*,
Ewl_Widget * *w*)

6.3.5.3 void ewl_box_cb_child_resize (Ewl_Container * *c*, Ewl_Widget * *w*, int
size, Ewl_Orientation *o*)

6.3.5.4 void ewl_box_cb_child_show (Ewl_Container * *c*, Ewl_Widget * *w*)

6.3.5.5 void ewl_box_cb_configure (Ewl_Widget * *w*, void * *ev_data*, void *
user_data)

6.3.5.6 void ewl_box_cb_configure_homogeneous (Ewl_Widget * *w*, void *
ev_data, void * *user_data*)

6.3.5.7 void ewl_box_homogeneous_set (Ewl_Box * *b*, unsigned int *h*)

Change the homogeneous layout of the box.

Parameters:

b,: the box to change homogeneous layout

h,: the boolean value to change the layout mode to

Returns:

Returns no value.

Boxes use homogeneous layout by default, this can be used to change that.

6.3.5.8 int ewl_box_init (Ewl_Box * *b*)

Initialize the box to starting values.

Parameters:

b,: the box to initialize

Returns:

Returns no value.

Responsible for setting up default values and callbacks within a box structure.

6.3.5.9 Ewl_Widget* ewl_box_new (void)

Allocate and initialize a new box with given orientation.

Returns:

Returns NULL on failure, or a newly allocated box on success.

6.3.5.10 Ewl_Orientation ewl_box_orientation_get (Ewl_Box * b)

Retrieves the orientation of the box.

Parameters:

b,: the box to retrieve orientation

Returns:

Returns the orientation value of the box *b*.

6.3.5.11 void ewl_box_orientation_set (Ewl_Box * b, Ewl_Orientation o)

Change the specified box's orientation.

Parameters:

b,: the box to change the orientation

o,: the orientation to set for the box

Returns:

Returns no value.

Changes the orientation of the specified box, and reconfigures it in order for the appearance to be updated.

6.3.5.12 void ewl_box_spacing_set (Ewl_Box * b, int s)

Changes the spacing between the objects in the box.

Parameters:

b,: the box to change the spacing

s,: the spacing to put between the child widgets

Returns:

Returns no value.

Adjust the spacing of the specified box and reconfigure it to change the appearance.

6.3.5.13 Ewl_Widget* ewl_hbox_new (void)

Allocate and initialize a new box with horizontal orientation.

Returns:

Returns NULL on failure, or a newly allocated horizontal box on success.

6.3.5.14 Ewl_Widget* ewl_vbox_new (void)

Allocate and initialize a new box with vertical orientation.

Returns:

Returns NULL on failure, or a newly allocated vertical box on success.

6.4 Ewl_Button: The Basic Button

6.4.1 Detailed Description

The button class is a basic button with a label. This class inherits from the **Ewl_Box** (p. 586) to allow for placing any other widget inside the button.

Remarks:

Inherits from **Ewl_Stock** (p. 750).

6.4.2 Tutorial

The **Ewl_Button** (p. 589) includes an icon and label by default but is actually a full container that can hold any type of non-toplevel widget. Because of this flexibility many other widget types inherit from the button.

```
Ewl_Widget *button;
button = ewl_button_new();
ewl_button_label_set(EWL_BUTTON(button), "Button label");
ewl_widget_show(button);
```

Data Structures

- struct **Ewl_Button**

*A simple **Ewl_Widget** (p. 785) to provide for a clickable button in the UI.*

Defines

- #define **EWL_BUTTON**(button) ((**Ewl_Button** *) button)
- #define **EWL_BUTTON_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_BUTTON_TYPE))
- #define **EWL_BUTTON_TYPE** "button"

Typedefs

- typedef **Ewl_Button** **Ewl_Button**

Functions

- unsigned int **ewl_button_alignment_get** (**Ewl_Button** *b)
Gets the alignment of the contents of the button.
- void **ewl_button_alignment_set** (**Ewl_Button** *b, unsigned int align)
Set the alignment of the contents of the button.
- void **ewl_button_cb_key_down** (**Ewl_Widget** *w, void *ev, void *data)
- unsigned int **ewl_button_fill_policy_get** (**Ewl_Button** *b)
Retrieves the current file policy set on the button.

- void **ewl_button_fill_policy_set** (**Ewl_Button** *b, unsigned int fill)
Set the fill policy of the contents of the button.
- const char * **ewl_button_image_get** (**Ewl_Button** *b)
Get the image file used by the button.
- void **ewl_button_image_set** (**Ewl_Button** *b, const char *file, const char *key)
*Set the image file **file** and the image key **key** to use.*
- void **ewl_button_image_size_get** (**Ewl_Button** *b, int *width, int *height)
Get the size of the image inside the button.
- void **ewl_button_image_size_set** (**Ewl_Button** *b, int width, int height)
Set the size of the image inside the button.
- int **ewl_button_init** (**Ewl_Button** *b)
Initialize a button to starting values.
- const char * **ewl_button_label_get** (**Ewl_Button** *b)
Retrieve the label of the specified button.
- void **ewl_button_label_set** (**Ewl_Button** *b, const char *l)
Change the label of the specified button.
- **Ewl_Widget** * **ewl_button_new** (void)
Allocate and initialize a new button.
- **Ewl_View** * **ewl_button_view_get** (void)
*Creates and returns a view to be used by **Ewl_Button** (p. 589) widgets.*

6.4.3 Define Documentation

6.4.3.1 #define EWL_BUTTON(button) ((Ewl_Button *) button)

Typecast a pointer to an **Ewl_Button** (p. 589) pointer.

6.4.3.2 #define EWL_BUTTON_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_BUTTON_TYPE))

Returns TRUE if the widget is an **Ewl_Button** (p. 589), FALSE otherwise

6.4.3.3 #define EWL_BUTTON_TYPE "button"

The type name for the **Ewl_Button** (p. 589) widget

6.4.4 Typedef Documentation

6.4.4.1 typedef struct Ewl_Button Ewl_Button

The button provides a simple wrapper for creating a clickable **Ewl_Widget** (p. 785) with an **Ewl_Text** (p. 757) displayed inside.

6.4.5 Function Documentation

6.4.5.1 unsigned int ewl_button_alignment_get (Ewl_Button * *b*)

Gets the alignment of the contents of the button.

Parameters:

b,: The button to get the content alignment

Returns:

Returns no value.

6.4.5.2 void ewl_button_alignment_set (Ewl_Button * *b*, unsigned int *align*)

Set the alignment of the contents of the button.

Parameters:

b,: The button to set the content alignment

align,: The new alignment for button contents

Returns:

Returns no value.

6.4.5.3 void ewl_button_cb_key_down (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.4.5.4 unsigned int ewl_button_fill_policy_get (Ewl_Button * *b*)

Retrieves the current file policy set on the button.

Parameters:

b,: The button to get the content fill policy from

Returns:

Returns the current fill policy set on the button

6.4.5.5 void ewl_button_fill_policy_set (Ewl_Button * *b*, unsigned int *fill*)

Set the fill policy of the contents of the button.

Parameters:

- b*,: The button to set the content fill policy
- fill*,: The new fill policy for button contents

Returns:

Returns no value.

6.4.5.6 const char* ewl_button_image_get (Ewl_Button * *b*)

Get the image file used by the button.

Parameters:

- b*,: The button to the the image file from

Returns:

Returns the image file used in the button or NULL on failure

6.4.5.7 void ewl_button_image_set (Ewl_Button * *b*, const char * *file*, const char * *key*)

Set the image file *file* and the image key *key* to use.

Parameters:

- b*,: The button to set the image on
- file*,: The file to use for the image
- key*,: The edge key to use for the image (or NULL if not using edge)

Returns:

Returns no value.

6.4.5.8 void ewl_button_image_size_get (Ewl_Button * *b*, int * *width*, int * *height*)

Get the size of the image inside the button.

Parameters:

- b*,: The button to get the image size on
- width*,: Where to put the width of the widget
- height*,: Where to put the height of the widget

Returns:

Returns no value.

6.4.5.9 void ewl_button_image_size_set (Ewl_Button * *b*, int *width*, int *height*)

Set the size of the image inside the button.

Parameters:

b,: The button to set the image size on

width,: The width of the image

height,: The height of the image

Returns:

Returns no value.

6.4.5.10 int ewl_button_init (Ewl_Button * *b*)

Initialize a button to starting values.

Parameters:

b,: the button to initialize

Returns:

Returns no value.

Initializes a button to default values and callbacks.

6.4.5.11 const char* ewl_button_label_get (Ewl_Button * *b*)

Retrieve the label of the specified button.

Parameters:

b,: the buttons whose label will be returned

Returns:

A newly allocated copy of the label on the button.

6.4.5.12 void ewl_button_label_set (Ewl_Button * *b*, const char * *l*)

Change the label of the specified button.

Parameters:

b,: the buttons whose label will be changed

l,: the new label for the button

Returns:

Returns no value.

6.4.5.13 Ewl_Widget* ewl_button_new (void)

Allocate and initialize a new button.

Returns:

Returns NULL on failure, a pointer to a new button on success

6.4.5.14 Ewl_View* ewl_button_view_get (void)

Creates and returns a view to be used by **Ewl_Button** (p. 589) widgets.

Returns:

Returns a view that can be used to display **Ewl_Button** (p. 589) widgets

6.5 Ewl_Calendar: The EWL Calendar Widget

6.5.1 Detailed Description

Defines the **Ewl_Calendar** (p. 591) class.

Remarks:

Inherits from **Ewl_Box** (p. 586).

Data Structures

- struct **Ewl_Calendar**

*Inherits from an **Ewl_Box** (p. 586) to provide a calendar widget.*

Defines

- #define **EWL_CALENDAR**(calendar) ((**Ewl_Calendar** *) calendar)
- #define **EWL_CALENDAR_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_CALENDAR_TYPE**))
- #define **EWL_CALENDAR_TYPE** "calendar"

Typedefs

- typedef **Ewl_Calendar** **Ewl_Calendar**

Functions

- char * **ewl_calendar_ascii_time_get** (**Ewl_Calendar** *cal)
Returns an ASCII formatted representation of the selected date the user must free this string.
- int **ewl_calendar_day_get** (**Ewl_Calendar** *c)
Returns the current day selected in the calendar.
- int **ewl_calendar_init** (**Ewl_Calendar** *calendar)
*Init a new **Ewl_Calendar** (p. 591) to default values and callbacks, and set date to today.*
- int **ewl_calendar_month_get** (**Ewl_Calendar** *c)
Returns the month selected in the calendar.
- **Ewl_Widget** * **ewl_calendar_new** (void)
*Creates a new **Ewl_Calendar** (p. 591).*
- int **ewl_calendar_year_get** (**Ewl_Calendar** *c)
Returns the year currently selected in the calendar.

6.5.2 Define Documentation

6.5.2.1 `#define EWL_CALENDAR(calendar) ((Ewl_Calendar *) calendar)`

Typecast a pointer to an **Ewl_Calendar** (p. 591) pointer

6.5.2.2 `#define EWL_CALENDAR_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_CALENDAR_TYPE))`

Returns TRUE if the widget is an **Ewl_Calendar** (p. 591), FALSE otherwise

6.5.2.3 `#define EWL_CALENDAR_TYPE "calendar"`

The type name for the **Ewl_Calendar** (p. 591) widget

6.5.3 Typedef Documentation

6.5.3.1 `typedef struct Ewl_Calendar Ewl_Calendar`

The **Ewl_Calendar** (p. 591) structure

6.5.4 Function Documentation

6.5.4.1 `char* ewl_calendar_ascii_time_get (Ewl_Calendar * cal)`

Returns an ASCII formatted representation of the selected date the user must free this string.

Parameters:

cal,: The calendar to get the date from

Returns:

Returns the string representing the selected date. This string must be free'd

6.5.4.2 `int ewl_calendar_day_get (Ewl_Calendar * c)`

Returns the current day selected in the calendar.

Parameters:

c,: The **Ewl_Calendar** (p. 591) to get the day from

Returns:

Returns the day currently selected in the calendar

6.5.4.3 int ewl_calendar_init (Ewl_Calendar * *ib*)

Init a new **Ewl_Calendar** (p. 591) to default values and callbacks, and set date to today.

Parameters:

ib,: The calendar widget to initialize

Returns:

Returns FALSE on failure, a TRUE on success

6.5.4.4 int ewl_calendar_month_get (Ewl_Calendar * *c*)

Returns the month selected in the calendar.

Parameters:

c,: The **Ewl_Calendar** (p. 591) to get the month from

Returns:

Returns the month currently selected in the calendar

6.5.4.5 Ewl_Widget* ewl_calendar_new (void)

Creates a new **Ewl_Calendar** (p. 591).

Returns:

Returns NULL on failure, a new **Ewl_Calendar** (p. 591) on success

Creates a new **Ewl_Calendar** (p. 591) object

6.5.4.6 int ewl_calendar_year_get (Ewl_Calendar * *c*)

Returns the year currently selected in the calendar.

Parameters:

c,: The **Ewl_Calendar** (p. 591) to get the year from

Returns:

Returns the current year selected in the calendar

6.6 Ewl_Callback: The Callback Mechanisms

6.6.1 Detailed Description

Defines methods for creating and modifying callbacks on widgets

Data Structures

- struct **Ewl_Callback**
Contains information about a callback.
- struct **Ewl_Callback_Custom**
Contains information about custom callbacks.

Defines

- #define **EWL_CALLBACK**(callback) ((**Ewl_Callback** *) callback)
- #define **EWL_CALLBACK_CUSTOM**(cb) ((**Ewl_Callback_Custom** *)cb)
- #define **EWL_CALLBACK_FLAG_INTERCEPT**(w, t) w → callbacks[EWL_CALLBACK_INDEX(t)].mask |= EWL_CALLBACK_NOTIFY_INTERCEPT
- #define **EWL_CALLBACK_FLAG_NOINTERCEPT**(w, t)
- #define **EWL_CALLBACK_FLAG_NOTIFY**(w, t) w → callbacks[EWL_CALLBACK_INDEX(t)].mask |= EWL_CALLBACK_NOTIFY_NOTIFY
- #define **EWL_CALLBACK_FLAGS**(w, t) (w → callbacks[EWL_CALLBACK_INDEX(t)].mask)
- #define **EWL_CALLBACK_FUNCTION**(cb_func) ((**Ewl_Callback_Function**) cb_func)
- #define **EWL_CALLBACK_INDEX**(t) ((t < EWL_CALLBACK_MAX) ? t : EWL_CALLBACK_MAX)
- #define **EWL_CALLBACK_LEN**(w, t) (w → callbacks[EWL_CALLBACK_INDEX(t)].len)
- #define **EWL_CALLBACK_LIST**(w, t) (w → callbacks[EWL_CALLBACK_INDEX(t)].list)
- #define **EWL_CALLBACK_NOTIFY_MASK** (0x3)
- #define **EWL_CALLBACK_POS**(w, t) w → callbacks[EWL_CALLBACK_INDEX(t)].index
- #define **EWL_CALLBACK_SET_DIRECT**(w, t) w → callbacks[EWL_CALLBACK_INDEX(t)].mask |= EWL_CALLBACK_TYPE_DIRECT
- #define **EWL_CALLBACK_SET_NODIRECT**(w, t) w → callbacks[EWL_CALLBACK_INDEX(t)].mask &= ~EWL_CALLBACK_TYPE_DIRECT

Typedefs

- typedef **Ewl_Callback** **Ewl_Callback**
- typedef **Ewl_Callback_Custom** **Ewl_Callback_Custom**
- typedef void(*) **Ewl_Callback_Function** (**Ewl_Widget** *widget, void *ev_data, void *user_data)

Functions

- **int ewl_callback_append** (**Ewl_Widget** *widget, unsigned int type, **Ewl_Callback_Function** func, void *user_data)
Append a callback of the specified type.
- **void ewl_callback_call** (**Ewl_Widget** *widget, unsigned int type)
Execute callbacks of specified types for the widget.
- **void ewl_callback_call_with_event_data** (**Ewl_Widget** *widget, unsigned int type, void *event_data)
Execute callbacks with event data.
- **void ewl_callback_clear** (**Ewl_Widget** *widget)
- **void ewl_callback_del** (**Ewl_Widget** *w, unsigned int t, **Ewl_Callback_Function** f)
Delete the specified callback function from the widget.
- **void ewl_callback_del_type** (**Ewl_Widget** *w, unsigned int t)
- **void ewl_callback_del_with_data** (**Ewl_Widget** *w, unsigned int t, **Ewl_Callback_Function** f, void *data)
Delete the specified callback function from the widget.
- **int ewl_callback_insert_after** (**Ewl_Widget** *w, unsigned int t, **Ewl_Callback_Function** f, void *user_data, **Ewl_Callback_Function** after, void *after_data)
Add a callback after a previous callback in list.
- **int ewl_callback_prepend** (**Ewl_Widget** *widget, unsigned int type, **Ewl_Callback_Function** func, void *user_data)
prepend a callback of the specified type
- **unsigned int ewl_callback_type_add** (void)
Creates and returns a new callback identifier.
- **int ewl_callbacks_init** (void)
Setup internal registration variables for callbacks.
- **void ewl_callbacks_shutdown** (void)
Destroy internal registration variables for callbacks.

6.6.2 Define Documentation

6.6.2.1 #define EWL_CALLBACK(callback) ((Ewl_Callback *) callback)

Typecasts a pointer to an **Ewl_Callback** (p. 593) pointer.

6.6.2.2 #define EWL_CALLBACK_CUSTOM(cb) ((Ewl_Callback_Custom *)cb)

Typecasts a pointer to an **Ewl_Callback_Custom** (p. 595) pointer.

6.6.2.3 `#define EWL_CALLBACK_FLAG_INTERCEPT(w, t) w → callbacks[EWL_CALLBACK_INDEX(t)].mask |= EWL_CALLBACK_NOTIFY_INTERCEPT`

Sets the callback intercept flag from a widget for a certain event type.

6.6.2.4 `#define EWL_CALLBACK_FLAG_NOINTERCEPT(w, t)`

Value:

```
w->callbacks[EWL_CALLBACK_INDEX(t)].mask = \
    w->callbacks[EWL_CALLBACK_INDEX(t)].mask & ~EWL_CALLBACK_NOTIFY_INTERCEPT
```

Clears the callback intercept flag from a widget for a certain event type.

6.6.2.5 `#define EWL_CALLBACK_FLAG_NOTIFY(w, t) w → callbacks[EWL_CALLBACK_INDEX(t)].mask |= EWL_CALLBACK_NOTIFY_NOTIFY`

Sets the callback notify flag from a widget for a certain event type.

6.6.2.6 `#define EWL_CALLBACK_FLAGS(w, t) (w → callbacks[EWL_CALLBACK_INDEX(t)].mask)`

Retrieves the callback flags from a widget for a certain event type.

6.6.2.7 `#define EWL_CALLBACK_FUNCTION(cb_func) ((Ewl_Callback_Function) cb_func)`

Typecasts a pointer to an `Ewl_Callback_Function` pointer.

6.6.2.8 `#define EWL_CALLBACK_INDEX(t) ((t < EWL_CALLBACK_MAX) ? t : EWL_CALLBACK_MAX)`

Retrieves the actual array index for `t`

6.6.2.9 `#define EWL_CALLBACK_LEN(w, t) (w → callbacks[EWL_CALLBACK_INDEX(t)].len)`

Retrieves the length from a widget for a certain event type.

6.6.2.10 `#define EWL_CALLBACK_LIST(w, t) (w → callbacks[EWL_CALLBACK_INDEX(t)].list)`

Retrieves the callback list from a widget for a certain event type.

6.6.2.11 `#define EWL_CALLBACK_NOTIFY_MASK (0x3)`

The value to binary AND with the callback pointer to check the notifiers.

6.6.2.12 `#define EWL_CALLBACK_POS(w, t) w → callbacks[EWL_CALLBACK_INDEX(t)].index`

Retrives the current callback position from a widget for an event type.

6.6.2.13 `#define EWL_CALLBACK_SET_DIRECT(w, t) w
→ callbacks[EWL_CALLBACK_INDEX(t)].mask |=
EWL_CALLBACK_TYPE_DIRECT`

Sets the callback direct flag for a certain event type

6.6.2.14 `#define EWL_CALLBACK_SET_NODIRECT(w, t) w
→ callbacks[EWL_CALLBACK_INDEX(t)].mask &=
~EWL_CALLBACK_TYPE_DIRECT`

Clears the callback direct flag from a widget for a certain event type

6.6.3 Typedef Documentation

6.6.3.1 `typedef struct Ewl_Callback Ewl_Callback`

The callbacks used internally for tracking event actions.

6.6.3.2 `typedef struct Ewl_Callback_Custom Ewl_Callback_Custom`

The custom callbacks for extended events

6.6.3.3 `typedef void(*) Ewl_Callback_Function(Ewl_Widget *widget, void
*ev_data, void *user_data)`

A shortcut for declaring functions that take a callback function pointer.

6.6.4 Function Documentation

6.6.4.1 `int ewl_callback_append (Ewl_Widget *w, unsigned int t,
Ewl_Callback_Function f, void *user_data)`

Append a callback of the specified type.

Parameters:

- w*,: the widget to attach the callback
- t*,: the type of the callback that is being attached
- f*,: the function to attach as a callback
- user_data*,: the data to be passed to the callback function

Returns:

Returns 0 on failure, the id of the new callback on success.

Allocates a new callback for the specified widget that calls f with $user_data$ as the data parameter when event ta occurs. This event is placed at the end of the callback chain.

6.6.4.2 void ewl_callback_call (Ewl_Widget * w , unsigned int t)

Execute callbacks of specified types for the widget.

Parameters:

- w ,: the widget to execute the callbacks
- t ,: the type of the callbacks to be executed

Returns:

Returns no value.

Executes the callback chain for the specified widget w , with event t .

6.6.4.3 void ewl_callback_call_with_event_data (Ewl_Widget * w , unsigned int t , void * ev_data)

Execute callbacks with event data.

Parameters:

- w ,: the widget to execute the callbacks
- t ,: the type of the callbacks to be executed
- ev_data ,: the event data to pass to the callbacks

Returns:

Returns no value.

Similar to ewl_callback_call, but the event data is substituted by ev_data .

6.6.4.4 void ewl_callback_clear (Ewl_Widget * $widget$)

6.6.4.5 void ewl_callback_del (Ewl_Widget * w , unsigned int t , Ewl_Callback_Function f)

Delete the specified callback function from the widget.

Parameters:

- w ,: the widget to delete the callback
- t ,: the type of event associated with the callback
- f ,: the function called by the callback

Returns:

Returns no value. Delete and frees the callback that calls function f when event t occurs to widget w .

6.6.4.6 void ewl_callback_del_type (Ewl_Widget * *w*, unsigned int *t*)

6.6.4.7 void ewl_callback_del_with_data (Ewl_Widget * *w*, unsigned int *t*,
Ewl_Callback_Function *f*, void * *d*)

Delete the specified callback function from the widget.

Parameters:

w,: the widget to delete the callback
t,: the type of event associated with the callback
f,: the function called by the callback
d,: the data passed to the callback

Returns:

Returns no value. Delete and frees the callback that calls function *f* when event *t* occurs to widget *w*.

6.6.4.8 int ewl_callback_insert_after (Ewl_Widget * *w*, unsigned int *t*,
Ewl_Callback_Function *f*, void * *user_data*, Ewl_Callback_Function
after, void * *after_data*)

Add a callback after a previous callback in list.

Parameters:

w,: the widget to insert the callback
t,: the type of the callback that is being attached
f,: the function to attach as a callback
user_data,: the data to be passed to the callback function
after,: the function of the callback to append after
after_data,: the user data of the callback to append after

Returns:

Returns 0 on failure, the id of the new callback on success.

Same functionality as ewl_callback_append, but the callback is placed after the specified callback on the callback chain.

6.6.4.9 int ewl_callback_prepend (Ewl_Widget * *w*, unsigned int *t*,
Ewl_Callback_Function *f*, void * *user_data*)

prepend a callback of the specified type

Parameters:

w,: the widget to attach the callback
t,: the type of the callback that is being attached

f,: the function to attach as a callback

user_data,: the data to be passed to the callback function

Returns:

Returns 0 on failure, the id of the new callback on success.

Same functionality as `ewl_callback_append`, but the callback is placed at the beginning of the callback chain.

6.6.4.10 `unsigned int ewl_callback_type_add (void)`

Creates and returns a new callback identifier.

Returns:

Returns a new callback identifier

6.6.4.11 `int ewl_callbacks_init (void)`

Setup internal registration variables for callbacks.

Returns:

Returns no value.

Sets up some important variables for tracking callbacks that allow shared callbacks.

W/o shared callbacks `ewl_test` with all windows open has a top line of: 21279 ningerso 19 0 22972 22M 9412 R 6.0 8.0 0:40 `ewl_test` With shared callbacks `ewl_test` with all windows open has a top line of: 15901 ningerso 10 0 20120 19M 9148 S 0.0 7.0 0:34 `ewl_test`

So using shared callbacks saves us over 2 MB of memory in this case.

Ecore_list as the callback storage with all tests open has a top line of: 9121 dsinclai 15 0 71156 17m 4276 S 0.0 1.8 0:11.06 `ewl_test` Using an array as the callback storage with all tests open has a top line of: 21727 dsinclai 15 0 68360 15m 4304 S 0.0 1.5 0:09.73 `ewl_test`

So using an array for the callbacks saves us about 2MB of memory in this case.

6.6.4.12 `void ewl_callbacks_shutdown (void)`

Destroy internal registration variables for callbacks.

Returns:

Returns no value.

Destroys some important variables for tracking callbacks that allow shared callbacks.

6.7 Ewl_Cell: The Cell Container, Layout for a Single Widget

6.7.1 Detailed Description

Defines a container to layout a single child with all of it's available space.

Remarks:

Inherits from **Ewl_Container** (p. 618).

Data Structures

- struct **Ewl_Cell**

*Inherits from **Ewl_Container** (p. 618) for packing widgets inside a cell.*

Defines

- #define **EWL_CELL(t)** ((**Ewl_Cell** *)t)
- #define **EWL_CELL_IS(w)** (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_CELL_TYPE**))
- #define **EWL_CELL_TYPE** "cell"

Typedefs

- typedef **Ewl_Cell** **Ewl_Cell**

Functions

- void **ewl_cell_cb_child_resize** (**Ewl_Container** *c, **Ewl_Widget** *w, int size, **Ewl_Orientation** o)
- void **ewl_cell_cb_child_show** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_cell_cb_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- int **ewl_cell_init** (**Ewl_Cell** *cell)

Initialize the cell fields of an inheriting object.

- **Ewl_Widget** * **ewl_cell_new** (void)

Allocate and initialize a new cell.

6.7.2 Define Documentation

6.7.2.1 #define **EWL_CELL(t)** ((**Ewl_Cell** *)t)

Typecast a pointer to an **Ewl_Cell** (p. 596) pointer.

6.7.2.2 `#define EWL_CELL_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_CELL_TYPE))`

Returns TRUE if the widget is an **Ewl_Cell** (p. 596), FALSE otherwise

6.7.2.3 `#define EWL_CELL_TYPE "cell"`

The type name for the **Ewl_Cell** (p. 596) widget

6.7.3 Typedef Documentation

6.7.3.1 `typedef struct Ewl_Cell Ewl_Cell`

The **Ewl_Cell** (p. 596) widget

6.7.4 Function Documentation

6.7.4.1 `void ewl_cell_cb_child_resize (Ewl_Container * c, Ewl_Widget * w, int size, Ewl_Orientation o)`

6.7.4.2 `void ewl_cell_cb_child_show (Ewl_Container * c, Ewl_Widget * w)`

6.7.4.3 `void ewl_cell_cb_configure (Ewl_Widget * w, void * ev_data, void * user_data)`

6.7.4.4 `int ewl_cell_init (Ewl_Cell * cell)`

Initialize the cell fields of an inheriting object.

Parameters:

cell,: the cell object to initialize

Returns:

Returns TRUE on success, FALSE on failure.

The fields of the *cell* object are initialized to their defaults.

6.7.4.5 `Ewl_Widget* ewl_cell_new (void)`

Allocate and initialize a new cell.

Returns:

Returns a newly allocated cell on success, NULL on failure.

6.8 Ewl_Check: The Simple Check for a Checkbutton

6.8.1 Detailed Description

Defines a widget with single purpose of providing a button with a checked state, without any extra decorating.

Remarks:

Inherits from **Ewl_Widget** (p. 785).

Data Structures

- struct **Ewl_Check**

*Inherits from **Ewl_Widget** (p. 785) and extends to provide a checked state.*

Defines

- `#define EWL_CHECK(c) ((Ewl_Check *)c)`
- `#define EWL_CHECK_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_CHECK_TYPE))`
- `#define EWL_CHECK_TYPE "check"`

Typedefs

- `typedef Ewl_Check Ewl_Check`

Functions

- `void ewl_check_cb_clicked (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_check_cb_update_check (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_check_checked_set (Ewl_Check *c, int checked)`
Change the checked status of the check.
- `int ewl_check_init (Ewl_Check *c)`
Initialize the members and callbacks of a check.
- `int ewl_check_is_checked (Ewl_Check *c)`
Determine the check state of the check.
- `Ewl_Widget * ewl_check_new (void)`
Allocate and initialize a new check.

6.8.2 Define Documentation

6.8.2.1 `#define EWL_CHECK(c) ((Ewl_Check *)c)`

Typecasts a pointer to an **Ewl_Check** (p. 597) pointer.

6.8.2.2 `#define EWL_CHECK_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_CHECK_TYPE))`

Returns TRUE if the widget is an **Ewl_Check** (p. 597), FALSE otherwise

6.8.2.3 `#define EWL_CHECK_TYPE "check"`

The type name for the **Ewl_Check** (p. 597) widget

6.8.3 Typedef Documentation

6.8.3.1 `typedef struct Ewl_Check Ewl_Check`

A single purpose extension of **Ewl_Widget** (p. 785) to provide a checked state.

6.8.4 Function Documentation

6.8.4.1 `void ewl_check_cb_clicked (Ewl_Widget * w, void * ev_data, void * user_data)`

6.8.4.2 `void ewl_check_cb_update_check (Ewl_Widget * w, void * ev_data, void * user_data)`

6.8.4.3 `void ewl_check_checked_set (Ewl_Check * cb, int c)`

Change the checked status of the check.

Parameters:

cb,: the check to change the status

c,: the new status of the check

Returns:

Returns no value.

Changes the checked status of the check and updates it's appearance to reflect the change.

6.8.4.4 `int ewl_check_init (Ewl_Check * cb)`

Initialize the members and callbacks of a check.

Parameters:

cb,: the check to initialize

Returns:

Returns TRUE on success, FALSE on failure.

The internal structures and callbacks of the check are initialized ot default values.

6.8.4.5 int ewl_check_is_checked (Ewl_Check * *cb*)

Determine the check state of the check.

Parameters:

cb,: the check to examine for it's checked state

Returns:

Returns TRUE if the check is checked, FALSE if not.

6.8.4.6 Ewl_Widget* ewl_check_new (void)

Allocate and initialize a new check.

Returns:

Returns the newly allocated check on success, NULL on failure.

6.9 Ewl_Checkbutton: An Ewl_Checkbutton with Label

6.9.1 Detailed Description

Defines an **Ewl_Checkbutton** (p. 598) that inherits from **Ewl_Widget** (p. 785) and provides an **Ewl_Check** (p. 597) that changes value on each click.

Remarks:

Inherits from **Ewl_Button** (p. 589).

Data Structures

- struct **Ewl_Checkbutton**

*Inherits from **Ewl_Button** (p. 589) and expands to provide a stateful check button.*

Defines

- #define **EWL_CHECKBUTTON**(button) ((**Ewl_Checkbutton** *) button)
- #define **EWL_CHECKBUTTON_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_CHECKBUTTON_TYPE**))
- #define **EWL_CHECKBUTTON_TYPE** "checkbutton"

Typedefs

- typedef **Ewl_Checkbutton** **Ewl_Checkbutton**

Functions

- void **ewl_checkbutton_cb_clicked** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_checkbutton_checked_set** (**Ewl_Checkbutton** *cb, int checked)
Update the checked state of the check button.
- int **ewl_checkbutton_init** (**Ewl_Checkbutton** *cb)
Initialize the members and callbacks of a check button.
- int **ewl_checkbutton_is_checked** (**Ewl_Checkbutton** *cb)
Retrieve the checked state of a check button.
- void **ewl_checkbutton_label_position_set** (**Ewl_Checkbutton** *cb, **Ewl_Position** p)
Set the check buttons label position.
- **Ewl_Widget** * **ewl_checkbutton_new** (void)
Allocate and initialize a new check button.

6.9.2 Define Documentation

6.9.2.1 `#define EWL_CHECKBUTTON(button) ((Ewl_Checkbutton *) button)`

Typecasts a pointer to an **Ewl_Checkbutton** (p. 598) pointer.

6.9.2.2 `#define EWL_CHECKBUTTON_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_CHECKBUTTON_TYPE))`

Returns TRUE if the widget is an **Ewl_Checkbutton** (p. 598), FALSE otherwise

6.9.2.3 `#define EWL_CHECKBUTTON_TYPE "checkboxbutton"`

The type name for the **Ewl_Checkbutton** (p. 598) widget

6.9.3 Typedef Documentation

6.9.3.1 `typedef struct Ewl_Checkbutton Ewl_Checkbutton`

The button class is a basic button with a label. This class is inherited by the check button and radio button classes.

6.9.4 Function Documentation

6.9.4.1 `void ewl_checkboxbutton_cb_clicked (Ewl_Widget * w, void * ev_data, void * user_data)`

6.9.4.2 `void ewl_checkboxbutton_checked_set (Ewl_Checkbutton * cb, int c)`

Update the checked state of the check button.

Parameters:

- cb*,: The checkboxbutton to set the checked value of
- c*,: The checked value to set

Returns:

Returns no value.

6.9.4.3 `int ewl_checkboxbutton_init (Ewl_Checkbutton * cb)`

Initialize the members and callbacks of a check button.

Parameters:

- cb*,: the check button to initialize

Returns:

Returns no value.

The internal structures and callbacks of the checkboxbutton are initialized to default values.

6.9.4.4 `int ewl_checkbutton_is_checked (Ewl_Checkbutton * cb)`

Retrieve the checked state of a check button.

Parameters:

cb,: The checkbutton to work with

Returns:

Returns TRUE if the checkbutton is set, FALSE otherwise

6.9.4.5 `void ewl_checkbutton_label_position_set (Ewl_Checkbutton * cb,
Ewl_Position p)`

Set the check buttons label position.

Parameters:

cb,: the widget to change the label positioning

p,: the new position of the label

Returns:

Returns no value.

Changes the position of the label associated with the check button.

6.9.4.6 `Ewl_Widget* ewl_checkbutton_new (void)`

Allocate and initialize a new check button.

Returns:

Returns the newly allocated checkbutton on success, NULL on failure.

6.10 Ewl_Colordialog: A Simple Colour Dialog widget

6.10.1 Detailed Description

Remarks:

Inherits from **Ewl_Dialog** (p. 626).

Data Structures

- struct **Ewl_Colordialog**

*Inherit from the **Ewl_Dialog** (p. 626) to create a color dialog widget.*

Defines

- #define **EWL_COLORDIALOG**(cd) ((**Ewl_Colordialog** *) cd)
- #define **EWL_COLORDIALOG_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_COLORDIALOG_TYPE**))
- #define **EWL_COLORDIALOG_TYPE** "colordialog"

Typedefs

- typedef **Ewl_Colordialog** **Ewl_Colordialog**

Functions

- unsigned int **ewl_colordialog_alpha_get** (**Ewl_Colordialog** *cp)
This retrieves the current alpha value selected in the dialog.
- void **ewl_colordialog_alpha_set** (**Ewl_Colordialog** *cp, unsigned int alpha)
Sets the alpha value of the cd dialog to alpha.
- void **ewl_colordialog_cb_button_click** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_colordialog_cb_delete_window** (**Ewl_Widget** *w, void *ev, void *data)
- **Ewl_Color_Mode** **ewl_colordialog_color_mode_get** (**Ewl_Colordialog** *cp)
This will retrieve the current Ewl_Color_Mode set on the color dialog.
- void **ewl_colordialog_color_mode_set** (**Ewl_Colordialog** *cp, **Ewl_Color_Mode** type)
Sets the color dialog mode to type.
- void **ewl_colordialog_current_rgb_get** (**Ewl_Colordialog** *cp, unsigned int *r, unsigned int *g, unsigned int *b)
This will retrieve the rgb values from the color dialog.
- void **ewl_colordialog_current_rgb_set** (**Ewl_Colordialog** *cp, unsigned int r, unsigned int g, unsigned int b)
This sets the given rgb information into the color dialog.

- unsigned int **ewl_colordialog_has_alpha_get** (**Ewl_Colordialog** *cp)
This retrieves whether or not the color dialog is showing alpha information.
- void **ewl_colordialog_has_alpha_set** (**Ewl_Colordialog** *cp, unsigned int alpha)
This will set if the color dialog will show the alpha information.
- int **ewl_colordialog_init** (**Ewl_Colordialog** *cd)
Initialize a color dialog to starting values.
- **Ewl_Widget** * **ewl_colordialog_new** (void)
*Create a new **Ewl_Colordialog** (p. 601) widget.*
- void **ewl_colordialog_previous_rgb_get** (**Ewl_Colordialog** *cp, unsigned int *r, unsigned int *g, unsigned int *b)
This will retrieve the value current set into the previous color box of the color dialog rgb values from the color dialog.
- void **ewl_colordialog_previous_rgb_set** (**Ewl_Colordialog** *cp, unsigned int r, unsigned int g, unsigned int b)
This sets the given rgb values as the color to display in the previous color box of the color dialog.

6.10.2 Define Documentation

6.10.2.1 #define EWL_COLORDIALOG(cd) ((Ewl_Colordialog *) cd)

Typecast a pointer to an **Ewl_Colordialog** (p. 601) pointer.

6.10.2.2 #define EWL_COLORDIALOG_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_COLORDIALOG_TYPE))

Returns TRUE if the widget is an **Ewl_Colordialog** (p. 601), FALSE otherwise

6.10.2.3 #define EWL_COLORDIALOG_TYPE "colordialog"

The type name for the **Ewl_Colordialog** (p. 601) widget

6.10.3 Typedef Documentation

6.10.3.1 typedef struct Ewl_Colordialog Ewl_Colordialog

The **Ewl_Colordialog** (p. 601)

6.10.4 Function Documentation

6.10.4.1 unsigned int ewl_colordialog_alpha_get (Ewl_Colordialog * cd)

This retrieves the current alpha value selected in the dialog.

Parameters:

cd,: The color dialog to get the alpha value from

Returns:

Returns the current alpha value selected in the dialog

6.10.4.2 void ewl_colordialog_alpha_set (Ewl_Colordialog * *cd*, unsigned int *alpha*)

Sets the alpha value of the *cd* dialog to *alpha*.

Parameters:

cd,: The color dialog to set the alpha into

alpha,: The alpha value to set into the color dialog

Returns:

Returns no value

6.10.4.3 void ewl_colordialog_cb_button_click (Ewl_Widget * *w*, void * *ev*, void * *data*)**6.10.4.4 void ewl_colordialog_cb_delete_window (Ewl_Widget * *w*, void * *ev*, void * *data*)****6.10.4.5 Ewl_Color_Mode ewl_colordialog_color_mode_get (Ewl_Colordialog * *cd*)**

This will retrieve the current Ewl_Color_Mode set on the color dialog.

Parameters:

cd,: The color dialog to get the mode from

Returns:

Returns the current mode of the color dialog

6.10.4.6 void ewl_colordialog_color_mode_set (Ewl_Colordialog * *cd*, Ewl_Color_Mode *type*)

Sets the color dialog mode to *type*.

Parameters:

cd,: The color dialog to set the mode of

type,: The mode to set on the color dialog

Returns:

Returns no value.

6.10.4.7 void ewl_colordialog_current_rgb_get (Ewl_Colordialog * *cd*, unsigned int * *r*, unsigned int * *g*, unsigned int * *b*)

This will retrieve the rgb values from the color dialog.

Parameters:

cd,: The color dialog to get the rgb information from
r,: Where to store the red value
g,: Where to store the green value
b,: Where to store the blue value

Returns:

Returns no value.

6.10.4.8 void ewl_colordialog_current_rgb_set (Ewl_Colordialog * *cd*, unsigned int *r*, unsigned int *g*, unsigned int *b*)

This sets the given rgb information into the color dialog.

Parameters:

cd,: The color dialog to set the rgb values into
r,: The red value to set
g,: The green value to set
b,: The blue value to set

Returns:

Returns no value.

6.10.4.9 unsigned int ewl_colordialog_has_alpha_get (Ewl_Colordialog * *cd*)

This retrieves whether or not the color dialog is showing alpha information.

Parameters:

cd,: The color dialog to get the alpha information from

Returns:

Returns TRUE if the color dialog is showing alpha information

6.10.4.10 void ewl_colordialog_has_alpha_set (Ewl_Colordialog * *cd*, unsigned int *alpha*)

This will set if the color dialog will show the alpha information.

Parameters:

cd,: The color dialog to set the alpha value on

alpha,: If the color dialog should show the alpha option

Returns:

Returns no value

6.10.4.11 int ewl_colordialog_init (Ewl_Colordialog * *cd*)

Initialize a color dialog to starting values.

Parameters:

cd,: the color dialog to initialize

Returns:

Returns TRUE on success, FALSE on failure.

6.10.4.12 Ewl_Widget* ewl_colordialog_new (void)

Create a new **Ewl_Colordialog** (p. 601) widget.

Returns:

Returns a new **Ewl_Colordialog** (p. 601) widget

**6.10.4.13 void ewl_colordialog_previous_rgb_get (Ewl_Colordialog * *cd*,
unsigned int * *r*, unsigned int * *g*, unsigned int * *b*)**

This will retrieve the value current set into the previous color box of the color dialog rgb values from the color dialog.

Parameters:

cd,: The color dialog to get the rgb information from

r,: Where to store the red value

g,: Where to store the green value

b,: Where to store the blue value

Returns:

Returns no value.

**6.10.4.14 void ewl_colordialog_previous_rgb_set (Ewl_Colordialog * *cd*,
unsigned int *r*, unsigned int *g*, unsigned int *b*)**

This sets the given rgb values as the color to display in the previous color box of the color dialog.

Parameters:

cd,: The color dialog to set the previous rgb values into

r,: The red value to set

g,: The green value to set

b,: The blue value to set

Returns:

Returns no value.

6.11 Ewl_Colorpicker: The colour picking widget

6.11.1 Detailed Description

Remarks:

Inherits from **Ewl_Box** (p. 586).

6.11.2 Tutorial

The **Ewl_Colorpicker** (p. 603) widget provides a way to let the user select a color either by manually entering its values, or graphically via the built-in **Ewl_Spectrum** (p. 743) widget, which also provides a preview of the color. The color can be selected in RGB or HSV format (but can be retrieved only in RGB format for now).

```
Ewl_Widget *cp;
cp = ewl_colorpicker_new();
ewl_widget_show(cp);
```

Once you've setup the colour picker, you can read its current and previous value and set them. For example the following swaps the two.

```
unsigned int r,g,b;
ewl_colorpicker_current_rgb_get(EWL_COLORPICKER(cp), &r, &g, &b);
ewl_colorpicker_previous_rgb_set(EWL_COLORPICKER(cp), r, g, b);
```

You can also change the display mode of the spectrum programmatically

```
void ewl_colorpicker_color_mode_set(Ewl_Colorpicker *cp, Ewl_Color_Mode type);
Ewl_Color_Mode ewl_colorpicker_color_mode_get(Ewl_Colorpicker *cp);
```

Data Structures

- struct **Ewl_Colorpicker**

*Inherits from **Ewl_Box** (p. 586) to produce a color picker widget.*

Defines

- #define **EWL_COLORPICKER(cp)** ((Ewl_Colorpicker *) cp)
- #define **EWL_COLORPICKER_IS(w)** (ewl_widget_type_is(EWL_WIDGET(w), EWL_COLORPICKER_TYPE))
- #define **EWL_COLORPICKER_TYPE** "colorpicker"

Typedefs

- typedef **Ewl_Colorpicker** **Ewl_Colorpicker**

Functions

- unsigned int **ewl_colorpicker_alpha_get** (Ewl_Colorpicker *cp)
Retrive the alpha value from the color picker.
- void **ewl_colorpicker_alpha_set** (Ewl_Colorpicker *cp, unsigned int alpha)
Set the alpha value into the color picker.
- void **ewl_colorpicker_cb_dnd_data** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_colorpicker_cb_previous_clicked** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_colorpicker_cb_radio_change** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_colorpicker_cb_spinner_change** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_colorpicker_cb_square_change** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_colorpicker_cb_vertical_change** (Ewl_Widget *w, void *ev, void *data)
- Ewl_Color_Mode **ewl_colorpicker_color_mode_get** (Ewl_Colorpicker *cp)
Get the color mode of the color picker.
- void **ewl_colorpicker_color_mode_set** (Ewl_Colorpicker *cp, Ewl_Color_Mode type)
Set the color mode of the color picker.
- void **ewl_colorpicker_current_rgb_get** (Ewl_Colorpicker *cp, unsigned int *r, unsigned int *g, unsigned int *b)
Retrieve the current RGB values from the color picker.
- void **ewl_colorpicker_current_rgb_set** (Ewl_Colorpicker *cp, unsigned int r, unsigned int g, unsigned int b)
Set the current RGB values into the color picker.
- unsigned int **ewl_colorpicker_has_alpha_get** (Ewl_Colorpicker *cp)
Retrieves if the color picker displays alpha information.
- void **ewl_colorpicker_has_alpha_set** (Ewl_Colorpicker *cp, unsigned int alpha)
Set if the color picker displays alpha information.
- int **ewl_colorpicker_init** (Ewl_Colorpicker *cp)
Initializes the cp color picker widget.
- Ewl_Widget * **ewl_colorpicker_new** (void)
*Creates a new **Ewl_Colorpicker** (p. 603) widget.*
- void **ewl_colorpicker_previous_rgb_get** (Ewl_Colorpicker *cp, unsigned int *r, unsigned int *g, unsigned int *b)
Retrieve the previous RGB values from the color picker.
- void **ewl_colorpicker_previous_rgb_set** (Ewl_Colorpicker *cp, unsigned int r, unsigned int g, unsigned int b)
Set the previous RGB values into the color picker.

6.11.3 Define Documentation

6.11.3.1 `#define EWL_COLORPICKER(cp) ((Ewl_Colorpicker *) cp)`

Typecast a pointer to an **Ewl_Colorpicker** (p. 603) pointer.

6.11.3.2 `#define EWL_COLORPICKER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_COLORPICKER_TYPE))`

Returns TRUE if the widget is an **Ewl_Colorpicker** (p. 603), FALSE otherwise

6.11.3.3 `#define EWL_COLORPICKER_TYPE "colorpicker"`

The type name for the **Ewl_Colorpicker** (p. 603) widget

6.11.4 Typedef Documentation

6.11.4.1 `typedef struct Ewl_Colorpicker Ewl_Colorpicker`

The **Ewl_Colorpicker** (p. 603)

6.11.5 Function Documentation

6.11.5.1 `unsigned int ewl_colorpicker_alpha_get (Ewl_Colorpicker * cp)`

Retrieve the alpha value from the color picker.

Parameters:

cp,: The colourpicker to get the alpha value from

Returns:

Returns the alpha value of the colorpicker (0-255)

6.11.5.2 `void ewl_colorpicker_alpha_set (Ewl_Colorpicker * cp, unsigned int alpha)`

Set the alpha value into the color picker.

Parameters:

cp,: the colourpicker to set the alpha value too

alpha,: The alpha value to set.

- 6.11.5.3 `void ewl_colorpicker_cb_dnd_data (Ewl_Widget * w, void * ev, void * data)`
- 6.11.5.4 `void ewl_colorpicker_cb_previous_clicked (Ewl_Widget * w, void * ev, void * data)`
- 6.11.5.5 `void ewl_colorpicker_cb_radio_change (Ewl_Widget * w, void * ev, void * data)`
- 6.11.5.6 `void ewl_colorpicker_cb_spinner_change (Ewl_Widget * w, void * ev, void * data)`
- 6.11.5.7 `void ewl_colorpicker_cb_square_change (Ewl_Widget * w, void * ev, void * data)`
- 6.11.5.8 `void ewl_colorpicker_cb_vertical_change (Ewl_Widget * w, void * ev, void * data)`
- 6.11.5.9 `Ewl_Color_Mode ewl_colorpicker_color_mode_get (Ewl_Colorpicker * cp)`

Get the color mode of the color picker.

Parameters:

cp,: The colorpicker to get the mode from

Returns:

Returns the Ewl_Color_Mode of the colorpicker

- 6.11.5.10 `void ewl_colorpicker_color_mode_set (Ewl_Colorpicker * cp, Ewl_Color_Mode mode)`

Set the color mode of the color picker.

Parameters:

cp,: The colorpicker to set the mode on

mode,: The mode to set on the colorpicker

Returns:

Returns no value.

- 6.11.5.11 `void ewl_colorpicker_current_rgb_get (Ewl_Colorpicker * cp, unsigned int * r, unsigned int * g, unsigned int * b)`

Retrieve the current RGB values from the color picker.

Parameters:

cp,: The colorpicker to get the current rgb values from

r,: Where to store the red value
g,: Where to store the green value
b,: Where to store the blue value

Returns:

Returns no value

6.11.5.12 `void ewl_colorpicker_current_rgb_set (Ewl_Colorpicker * cp, unsigned int r, unsigned int g, unsigned int b)`

Set the current RGB values into the color picker.

Parameters:

cp,: The colourpicker to set the current rgb value into
r,: The red value to set
g,: The green value to set
b,: The blue value to set

Returns:

Returns no value.

6.11.5.13 `unsigned int ewl_colorpicker_has_alpha_get (Ewl_Colorpicker * cp)`

Retrieves if the color picker displays alpha information.

Parameters:

cp,: The colour picker to get the has_alpha flag from

Returns:

Returns the value of the has_alpha flag (TRUE or FALSE)

6.11.5.14 `void ewl_colorpicker_has_alpha_set (Ewl_Colorpicker * cp, unsigned int alpha)`

Set if the color picker displays alpha information.

Parameters:

cp,: The colour picker to set the has_alpha flag on.
alpha,: Value to set the has_alpha param too. (TRUE or FALSE)

Returns:

Returns no value.

6.11.5.15 `int ewl_colorpicker_init (Ewl_Colorpicker * cp)`

Initializes the `cp` color picker widget.

Parameters:

cp,: The `Ewl_Colorpicker` (p. 603) to initialize

Returns:

Returns TRUE on success or FALSE on failure

6.11.5.16 `Ewl_Widget* ewl_colorpicker_new (void)`

Creates a new `Ewl_Colorpicker` (p. 603) widget.

Returns:

Returns a new `Ewl_Colorpicker` (p. 603) widget or NULL on failure

6.11.5.17 `void ewl_colorpicker_previous_rgb_get (Ewl_Colorpicker * cp,
unsigned int * r, unsigned int * g, unsigned int * b)`

Retrieve the previous RGB values from the color picker.

Parameters:

cp,: The colorpicker to get the previous rgb value from.

r,: Where to store the red value

g,: Where to store the green value

b,: Where to store the blue value

Returns:

Returns no value.

6.11.5.18 `void ewl_colorpicker_previous_rgb_set (Ewl_Colorpicker * cp,
unsigned int r, unsigned int g, unsigned int b)`

Set the previous RBG values into the color picker.

Parameters:

cp,: The colorpicker to set the previous rgb value into

r,: The red value to set

g,: The green value to set

b,: The blue value to set

Returns:

Returns no value.

6.12 Ewl_Combo: A Simple Combo Box

6.12.1 Detailed Description

Defines a combo box used internally. The contents on the box are not drawn outside of the Evas.

Remarks:

Inherits from **Ewl_MVC** (p. 705).

6.12.2 Tutorial

The **Ewl_Combo** (p. 610) widget is based on a Model/View/Controller design. Before you can use the combo you need to setup your starting data structure, your model and your view. Once everything is created if you want to change your data all you have to do is tell the combo that it's data is dirty and it will redisplay the combo box.

```
model = ewl_model_new();
ewl_model_data_fetch_set(model, combo_test_data_fetch);
ewl_model_data_count_set(model, combo_test_data_count_get);

view = ewl_view_new();
ewl_view_constructor_set(view, ewl_label_new);
ewl_view_assign_set(view, EWL_VIEW_ASSIGN(ewl_label_text_set));
ewl_view_header_fetch_set(view, combo_test_data_header_fetch);

combo = ewl_combo_new();
ewl_callback_append(combo, EWL_CALLBACK_VALUE_CHANGED,
                    combo_value_changed, NULL);
ewl_mvc_model_set(EWL_MVC(combo), model);
ewl_mvc_view_set(EWL_MVC(combo), view);
ewl_mvc_data_set(EWL_MVC(combo), data);
ewl_widget_show(combo);
```

If you have a custom widget you wish to display you can set your own functions into the view to draw your widget. In this case we just want a simple label displayed.

The data header is optional and will be displayed at the top of your combo. In the case where the combo is editable it will use the header as the normal display. In this case you are responsible for creating and populating the widget.

```
static Ewl_Widget *combo_test_data_header_fetch(void *data, int col);
static void *combo_test_data_fetch(void *data, unsigned int row,
                                   unsigned int col);
static int combo_test_data_count_get(void *data);
```

The three model functions are responsible for getting the information from your model as needed. Each time the combo needs another row of data it will call the data_fetch function. The col parameter is unused by the combo box. The count_get function is responsible for returning a count of the number of items in your data structure. Each of these three functions receive a void *data param. This is your data as set into the combo box so you shouldn't need to create a global pointer to the data.

Note:

If you set the combo to editable, with ewl_combo_editable set then instead of using the model/view to get the data we will query the view for the header. It is then up to the app to

do what they will with the header to make it 'editable'. In most cases, this will mean packing an entry into there with the value from the data. In this case you will need to attach a `EWL_CALLBACK_VALUE_CHANGED` callback to the entry and handle its value change yourself. The combo won't know about any changed values in the entry and will always have the value from the data.

Data Structures

- struct **Ewl_Combo**

*Inherits from the **Ewl_MVC** (p. 705) widget to provide a combo widget.*

- struct **Ewl_Combo_Cell**

*Inherits from **Ewl_Cell** (p. 596) and is used internally for the combo box.*

Defines

- `#define EWL_COMBO(combo) ((Ewl_Combo *) combo)`
- `#define EWL_COMBO_CELL(c) ((Ewl_Combo_Cell *) c)`
- `#define EWL_COMBO_CELL_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_COMBO_CELL_TYPE))`
- `#define EWL_COMBO_CELL_TYPE "combo_cell"`
- `#define EWL_COMBO_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_COMBO_TYPE))`
- `#define EWL_COMBO_TYPE "combo"`

Typedefs

- `typedef Ewl_Combo Ewl_Combo`
- `typedef Ewl_Combo_Cell Ewl_Combo_Cell`

Functions

- `void ewl_combo_cb_decrement_clicked (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_combo_cb_popup_hide (Ewl_Widget *w, void *ev, void *data)`
- `Ewl_Combo * ewl_combo_cell_combo_get (Ewl_Combo_Cell *c)`
- `void ewl_combo_cell_combo_set (Ewl_Combo_Cell *c, Ewl_Combo *combo)`
- `void * ewl_combo_cell_data_get (Ewl_Combo_Cell *c)`
- `void ewl_combo_cell_data_set (Ewl_Combo_Cell *c, void *mvc_data)`
- `int ewl_combo_cell_init (Ewl_Combo_Cell *c)`

Initializes a combo cell to default values.

- `Ewl_Model * ewl_combo_cell_model_get (Ewl_Combo_Cell *c)`
- `void ewl_combo_cell_model_set (Ewl_Combo_Cell *c, Ewl_Model *model)`
- `Ewl_Widget * ewl_combo_cell_new (void)`

Create a new combo cell.

- `unsigned int ewl_combo_editable_get (Ewl_Combo *combo)`

Retrieves the editable status of the combo.

- void **ewl_combo_editable_set** (**Ewl_Combo** *combo, unsigned int editable)
This will set if the displayed data in the combo is editable.
- int **ewl_combo_init** (**Ewl_Combo** *combo)
Initializes a combo to default values.
- **Ewl_Widget** * **ewl_combo_new** (void)
Create a new combo box.
- void **ewl_combo_popup_container_set** (**Ewl_Combo** *combo, **Ewl_Container** *c)
- unsigned int **ewl_combo_scrollable_get** (**Ewl_Combo** *combo)
Retrieves the scrollable status of the combo.
- void **ewl_combo_scrollable_set** (**Ewl_Combo** *combo, unsigned int scrollable)

6.12.3 Define Documentation

6.12.3.1 #define EWL_COMBO(combo) ((Ewl_Combo *) combo)

Typecasts a pointer to an **Ewl_Combo** (p. 610) pointer.

6.12.3.2 #define EWL_COMBO_CELL(c) ((Ewl_Combo_Cell *)c)

The type name for the **Ewl_Combo_Cell** (p. 612) widget

6.12.3.3 #define EWL_COMBO_CELL_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_COMBO_CELL_TYPE))

Returns TRUE if the widget is an **Ewl_Combo_Cell** (p. 612), FALSE otherwise

6.12.3.4 #define EWL_COMBO_CELL_TYPE "combo_cell"

6.12.3.5 #define EWL_COMBO_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_COMBO_TYPE))

Returns TRUE if the widget is an **Ewl_Combo** (p. 610), FALSE otherwise

6.12.3.6 #define EWL_COMBO_TYPE "combo"

The type name for the **Ewl_Combo** (p. 610) widget

6.12.4 Typedef Documentation

6.12.4.1 typedef struct Ewl_Combo Ewl_Combo

A simple internal combo box, it is limited to drawing within the current evas.

6.12.4.2 typedef struct Ewl_Combo_Cell Ewl_Combo_Cell

Typedef for the **Ewl_Combo_Cell** (p. 612) struct

6.12.5 Function Documentation

6.12.5.1 void ewl_combo_cb_decrement_clicked (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.12.5.2 void ewl_combo_cb_popup_hide (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.12.5.3 Ewl_Combo* ewl_combo_cell_combo_get (Ewl_Combo_Cell * *cell*)

Parameters:

cell,: The **Ewl_Combo** (p. 610) to use

Returns:

Returns the parent combo of the cell

6.12.5.4 void ewl_combo_cell_combo_set (Ewl_Combo_Cell * *cell*, Ewl_Combo * *combo*)

Parameters:

cell,: The **Ewl_Combo_Cell** (p. 612) to use

combo,: the parent combo widget

Returns:

Returns no value

6.12.5.5 void* ewl_combo_cell_data_get (Ewl_Combo_Cell * *cell*)

Parameters:

cell,: The **Ewl_Combo** (p. 610) to use

Returns:

Returns the model of the cell

6.12.5.6 void ewl_combo_cell_data_set (Ewl_Combo_Cell * *cell*, void * *mvc_data*)

Parameters:

cell,: The **Ewl_Combo_Cell** (p. 612) to use

mvc_data,: the data for the cell

Returns:

Returns no value

6.12.5.7 int ewl_combo_cell_init (Ewl_Combo_Cell * *cell*)

Initializes a combo cell to default values.

Parameters:

combo,: The Ewl_Combo_Cell (p. 612) to initialize

Returns:

Returns TRUE on success or FALSE on failure

6.12.5.8 Ewl_Model* ewl_combo_cell_model_get (Ewl_Combo_Cell * *cell*)**Parameters:**

cell,: The Ewl_Combo (p. 610) to use

Returns:

Returns the model of the cell

6.12.5.9 void ewl_combo_cell_model_set (Ewl_Combo_Cell * *cell*, Ewl_Model * *model*)**Parameters:**

cell,: The Ewl_Combo_Cell (p. 612) to use

model,: the model for the cell

Returns:

Returns no value

6.12.5.10 Ewl_Widget* ewl_combo_cell_new (void)

Create a new combo cell.

Returns:

Returns a pointer to a new combo cell on success, NULL on failure.

6.12.5.11 unsigned int ewl_combo_editable_get (Ewl_Combo * *combo*)

Retrieves the editable status of the combo.

Parameters:

combo,: The **Ewl_Combo** (p. 610) to use

Returns:

Returns the editable status of the combo

6.12.5.12 void ewl_combo_editable_set (Ewl_Combo * *combo*, unsigned int *editable*)

This will set if the displayed data in the combo is editable.

Parameters:

combo,: The **Ewl_Combo** (p. 610) to use

editable,: Set if the combo is editable or not

Returns:

Returns no value

If the editable flag is set to TRUE then the combo will always request the header as the item to display. If it is FALSE then the combo will use the selected widget as the display.

6.12.5.13 int ewl_combo_init (Ewl_Combo * *combo*)

Initializes a combo to default values.

Parameters:

combo,: The **Ewl_Combo** (p. 610) to initialize

Returns:

Returns TRUE on success or FALSE on failure

6.12.5.14 Ewl_Widget* ewl_combo_new (void)

Create a new combo box.

Returns:

Returns a pointer to a new combo on success, NULL on failure.

6.12.5.15 void ewl_combo_popup_container_set (Ewl_Combo * *combo*,
Ewl_Container * *c*)

Parameters:

combo,: The Ewl_Combo (p. 610) to use

editable,: The Container to use in the popup

Returns:

Returns no value

6.12.5.16 unsigned int ewl_combo_scrollable_get (Ewl_Combo * *combo*)

Retrieves the scrollable status of the combo.

Parameters:

combo,: The Ewl_Combo (p. 610) to use

Returns:

Returns the scrollable status of the combo

6.12.5.17 void ewl_combo_scrollable_set (Ewl_Combo * *combo*, unsigned int
scrollable)

Parameters:

combo,: The Ewl_Combo (p. 610) to use

editable,: Set if the combo is scrollable or not

Returns:

Returns no value

On true, this functions set the combo to use a scrollpane to view the widget inside of it. The maximal size of it is set by a theme defined value.

6.13 Ewl_Config: Functions for Manipulating Configuration Data

Data Structures

- **struct Ewl_Config**
Contains Ewl configuration data.

Typedefs

- **typedef Ewl_Config Ewl_Config**

Functions

- **void ewl_config_cache_init** (void)
Initializes the configuration cache.
- **int ewl_config_can_save_system** (Ewl_Config *cfg)
Determines if the user can write the system config file.
- **void ewl_config_color_get** (Ewl_Config *cfg, const char *k, int *r, int *g, int *b, int *a)
Retrieve color associated with a key.
- **void ewl_config_color_set** (Ewl_Config *cfg, const char *k, int r, int g, int b, int a, Ewl_State_Type state)
Set the value of key to the specified color.
- **void ewl_config_destroy** (Ewl_Config *cfg)
Destroys the given config structure.
- **float ewl_config_float_get** (Ewl_Config *cfg, const char *k)
Retrieve floating point value associated with a key.
- **void ewl_config_float_set** (Ewl_Config *cfg, const char *k, float v, Ewl_State_Type state)
Set the value of key to the specified float.
- **int ewl_config_init** (void)
- **int ewl_config_int_get** (Ewl_Config *cfg, const char *k)
Retrieve integer value associated with a key.
- **void ewl_config_int_set** (Ewl_Config *cfg, const char *k, int v, Ewl_State_Type state)
Set the value of key to the specified integer.
- **Ewl_Config * ewl_config_new** (const char *app_name)
*Creates the *Ewl_Config* (p. 614) file for the given application.*
- **void ewl_config_shutdown** (void)

- `const char * ewl_config_string_get (Ewl_Config *cfg, const char *k)`
Retrieve string value associated with a key.
- `void ewl_config_string_set (Ewl_Config *cfg, const char *k, const char *v, Ewl_State_Type state)`
set the value of key to the specified string
- `int ewl_config_system_save (Ewl_Config *cfg)`
Writes out the system and user data to the system config file.
- `int ewl_config_user_save (Ewl_Config *cfg)`
Writes out the user config to the users config file.

6.13.1 Typedef Documentation

6.13.1.1 typedef struct Ewl_Config Ewl_Config

The **Ewl_Config** (p. 614) structure

6.13.2 Function Documentation

6.13.2.1 void ewl_config_cache_init (void)

6.13.2.2 int ewl_config_can_save_system (Ewl_Config * cfg)

Determines if the user can write the system config file.

Parameters:

cfg,: The **Ewl_Config** (p. 614) to work with

Returns:

Returns TRUE if the user can write to the system conf file, FALSE otherwise

6.13.2.3 void ewl_config_color_get (Ewl_Config * cfg, const char * k, int * r, int * g, int * b, int * a)

Retrieve color associated with a key.

Parameters:

cfg,: The **Ewl_Config** (p. 614) to work with

k,: the key to search

r,: the red value that is associated with the key

g,: the green value that is associated with the key

b,: the blue value that is associated with the key

a,: the alpha value that is associated with the key

Returns:

Returns no value

6.13.2.4 `void ewl_config_color_set (Ewl_Config * cfg, const char * k, int r, int g, int b, int a, Ewl_State_Type state)`

Set the value of key to the specified color.

Parameters:

cfg,: The **Ewl_Config** (p. 614) to work with
k,: the key to set in the configuration database
r,: the red value that will be associated with the key
g,: the green value that will be associated with the key
b,: the blue value that will be associated with the key
a,: the alpha value that will be associated with the key
state,: The state of the key being set, persistent or transient

Returns:

Returns no value

Sets the color value associated with the key *k* to *v* in the configuration database.

6.13.2.5 `void ewl_config_destroy (Ewl_Config * cfg)`

Destroys the given config structure.

Parameters:

cfg,: The **Ewl_Config** (p. 614) to destroy

Returns:

Returns no value

6.13.2.6 `float ewl_config_float_get (Ewl_Config * cfg, const char * k)`

Retrieve floating point value associated with a key.

Parameters:

cfg,: The **Ewl_Config** (p. 614) to work with
k,: the key to search

Returns:

Returns the found float value on success, 0.0 on failure.

6.13.2.7 void ewl_config_float_set (Ewl_Config * *cfg*, const char * *k*, float *v*, Ewl_State_Type *state*)

Set the value of key to the specified float.

Parameters:

- cfg*,: The **Ewl_Config** (p.614) to work with
- k*,: the key to set in the configuration database
- v*,: the float value that will be associated with the key
- state*,: The state of the key being set, persistent or transient

Returns:

Returns no value

Sets the float value associated with the key *k* to *v* in the configuration database.

6.13.2.8 int ewl_config_init (void)**6.13.2.9 int ewl_config_int_get (Ewl_Config * *cfg*, const char * *k*)**

Retrieve integer value associated with a key.

Parameters:

- cfg*,: The **Ewl_Config** (p.614) to work with
- k*,: the key to search

Returns:

Returns the found integer value on success, 0 on failure.

6.13.2.10 void ewl_config_int_set (Ewl_Config * *cfg*, const char * *k*, int *v*, Ewl_State_Type *state*)

Set the value of key to the specified integer.

Parameters:

- cfg*,: The **Ewl_Config** (p.614) to work with
- k*,: the key to set in the configuration database
- v*,: the integer value that will be associated with the key
- state*,: The state of the key being set, persistent or transient

Returns:

Returns TRUE on success, FALSE on failure.

Sets the integer value associated with the key *k* to *v* in the configuration database.

6.13.2.11 Ewl_Config* ewl_config_new (const char * *app_name*)

Creates the **Ewl_Config** (p. 614) file for the given application.

Parameters:

app_name,: The name of the app to open the config for

Returns:

Returns the **Ewl_Config** (p. 614) struct for this app

6.13.2.12 void ewl_config_shutdown (void)**6.13.2.13 const char* ewl_config_string_get (Ewl_Config * *cfg*, const char * *k*)**

Retrieve string value associated with a key.

Parameters:

cfg,: The **Ewl_Config** (p. 614) to work with

k,: the key to search

Returns:

Returns the found string value on success, NULL on failure.

6.13.2.14 void ewl_config_string_set (Ewl_Config * *cfg*, const char * *k*, const char * *v*, Ewl_State_Type *state*)

set the value of key to the specified string

Parameters:

cfg,: The **Ewl_Config** (p. 614) to work with

k,: the key to set in the configuration database

v,: the string value that will be associated with the key

state,: The state of the key being set, persistent or transient

Returns:

Returns no value.

Sets the string value associated with the key *k* to *v* in the configuration database.

6.13.2.15 int ewl_config_system_save (Ewl_Config * *cfg*)

Writes out the system and user data to the system config file.

Parameters:

cfg,: The **Ewl_Config** (p. 614) to save

Returns:

Returns TRUE on success, FALSE on failure

6.13.2.16 `int ewl_config_user_save (Ewl_Config * cfg)`

Writes out the user config to the users config file.

Parameters:

cfg,: The **Ewl_Config** (p. 614) to work with

Returns:

Returns TRUE on success or FALSE on failure

6.14 Ewl_Container: Widgets Holding Other Widgets

6.14.1 Detailed Description

Define the **Ewl_Container** (p. 618) class which inherits from **Ewl_Widget** (p. 785) and adds the ability to nest Ewl_Widget's inside.

Remarks:

Inherits from **Ewl_Widget** (p. 785).

Data Structures

- struct **Ewl_Container**

*Inherits from the **Ewl_Widget** (p. 785) and expands to allow for placing child widgets within the available space. Also adds notifiers for various child events.*

Defines

- #define **EWL_CONTAINER(widget)** ((**Ewl_Container** *) widget)
*Typecast a pointer to an **Ewl_Container** (p. 618) pointer.*
- #define **EWL_CONTAINER_IS(w)** (ewl_widget_type_is(EWL_WIDGET(w), EWL_CONTAINER_TYPE))
- #define **EWL_CONTAINER_TYPE** "container"

Typedefs

- typedef void(*) **Ewl_Child_Add** (**Ewl_Container** *c, **Ewl_Widget** *w)
- typedef void(*) **Ewl_Child_Hide** (**Ewl_Container** *c, **Ewl_Widget** *w)
- typedef void(*) **Ewl_Child_Remove** (**Ewl_Container** *c, **Ewl_Widget** *w, int idx)
- typedef void(*) **Ewl_Child_Resize** (**Ewl_Container** *c, **Ewl_Widget** *w, int size, **Ewl_Orientation** o)
- typedef void(*) **Ewl_Child_Show** (**Ewl_Container** *c, **Ewl_Widget** *w)
- typedef **Ewl_Container** **Ewl_Container**
- typedef **Ewl_Widget** *(*) **Ewl_Container_Iterator** (**Ewl_Container** *c)

Functions

- void **ewl_container_add_notify_set** (**Ewl_Container** *container, **Ewl_Child_Add** add)
Set the function to be called when adding children.
- void **ewl_container_callback_intercept** (**Ewl_Container** *c, unsigned int t)
Receive a callback in place of children.
- void **ewl_container_callback_nointercept** (**Ewl_Container** *c, unsigned int t)
Remove a callback interception from children.

- void **ewl_container_callback_notify** (Ewl_Container *c, unsigned int t)
receive a callback of a child
- void **ewl_container_cb_configure** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_cb_disable** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_cb_enable** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_cb_obscure** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_cb_realize** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_cb_reparent** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_cb_reveal** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_cb_unrealize** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_child_add_call** (Ewl_Container *c, Ewl_Widget *w)
Triggers the child_add callback for the container c.
- void **ewl_container_child_append** (Ewl_Container *parent, Ewl_Widget *child)
Add a child at the end of the container.
- Ewl_Widget * **ewl_container_child_at_get** (Ewl_Container *widget, int x, int y)
Find the child at the given coordinates.
- Ewl_Widget * **ewl_container_child_at_recursive_get** (Ewl_Container *widget, int x, int y)
Find child that intersects coordinates.
- int **ewl_container_child_count_get** (Ewl_Container *c)
Returns the number of child widgets in the container.
- int **ewl_container_child_count_internal_get** (Ewl_Container *c)
Returns the number of child widgets in the container.
- Ewl_Widget * **ewl_container_child_get** (Ewl_Container *parent, int index)
Retrieve the child at the given index in the container.
- void **ewl_container_child_hide_call** (Ewl_Container *c, Ewl_Widget *w)
Triggers the child_hide callback for the container c.
- int **ewl_container_child_index_get** (Ewl_Container *parent, Ewl_Widget *child)
Retrieves the index of the given child in the container or -1 if not found.
- int **ewl_container_child_index_internal_get** (Ewl_Container *parent, Ewl_Widget *child)
Retrieves the index of the given child in the container taking internal widgets into account or -1 if not found.

- **void ewl_container_child_insert** (**Ewl_Container** *parent, **Ewl_Widget** *child, int index)
Add a child at an index of the container.
- **void ewl_container_child_insert_internal** (**Ewl_Container** *parent, **Ewl_Widget** *child, int index)
Add a child at an index of the container.
- **Ewl_Widget * ewl_container_child_internal_get** (**Ewl_Container** *parent, int index)
Retrieves the child at the given index in the container taking internal widgets into account.
- **void ewl_container_child_iterate_begin** (**Ewl_Container** *c)
Prepare the container to iterate over it's children.
- **void ewl_container_child_iterator_set** (**Ewl_Container** *c, **Ewl_Container_Iterator** i)
Changes the iterator function for a container.
- **Ewl_Widget * ewl_container_child_next** (**Ewl_Container** *c)
Retrieve the next elligible child in a container.
- **void ewl_container_child_prepend** (**Ewl_Container** *parent, **Ewl_Widget** *child)
Add a child at the beginning of the container.
- **void ewl_container_child_remove** (**Ewl_Container** *parent, **Ewl_Widget** *child)
Remove the specified child from the container.
- **void ewl_container_child_remove_call** (**Ewl_Container** *c, **Ewl_Widget** *w, int idx)
Triggers the child_remove callback for the container c.
- **void ewl_container_child_resize** (**Ewl_Widget** *w, int size, **Ewl_Orientation** o)
Notify a container of a child pref size change.
- **void ewl_container_child_show_call** (**Ewl_Container** *c, **Ewl_Widget** *w)
Triggers the child_show callback for the container c.
- **void ewl_container_destroy** (**Ewl_Container** *c)
Destroy all the sub-children of the container.
- **Ewl_Container * ewl_container_end_redirect_get** (**Ewl_Container** *c)
Searches for the last redirected container of the container.
- **void ewl_container_hide_notify_set** (**Ewl_Container** *container, **Ewl_Child_Hide** show)
Set the function to be called when hideing children.

- **int ewl_container_init** (**Ewl_Container** *container)
Initialize a containers default fields and callbacks.
- **void ewl_container_largest_prefer** (**Ewl_Container** *c, **Ewl_Orientation** o)
Set preferred size to widest child.
- **Ewl_Container * ewl_container_redirect_get** (**Ewl_Container** *c)
Retrieves for the redirected container of the container.
- **void ewl_container_redirect_set** (**Ewl_Container** *c, **Ewl_Container** *rc)
Changes the redirected container of the container.
- **void ewl_container_remove_notify_set** (**Ewl_Container** *container, **Ewl_Child_Remove** remove)
Set the function to be called when removing children.
- **void ewl_container_reset** (**Ewl_Container** *c)
Destroy all children of the container.
- **void ewl_container_resize_notify_set** (**Ewl_Container** *container, **Ewl_Child_Resize** resize)
Set function to be called when children resize.
- **void ewl_container_show_notify_set** (**Ewl_Container** *container, **Ewl_Child_Show** show)
Set the function to be called when showing children.
- **void ewl_container_sum_prefer** (**Ewl_Container** *c, **Ewl_Orientation** o)
Set the sum orientation of the container.

6.14.2 Define Documentation

6.14.2.1 #define EWL_CONTAINER(widget) ((Ewl_Container *) widget)

Typecast a pointer to an **Ewl_Container** (p. 618) pointer.

6.14.2.2 #define EWL_CONTAINER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_CONTAINER_TYPE))

Returns TRUE if the widget is an **Ewl_Container** (p. 618), FALSE otherwise

6.14.2.3 #define EWL_CONTAINER_TYPE "container"

The type name for the **Ewl_Container** (p. 618) widget

6.14.3 Typedef Documentation

6.14.3.1 `typedef void(*) Ewl_Child_Add(Ewl_Container *c, Ewl_Widget *w)`

A typedef to shorten the definition of the `child_add` callbacks. This callback is container specific and is triggered when an **Ewl_Widget** (p. 785) is added to the **Ewl_Container** (p. 618).

6.14.3.2 `typedef void(*) Ewl_Child_Hide(Ewl_Container *c, Ewl_Widget *w)`

A typedef to shorten the definition of the `child_hide` callbacks. This callback is container specific and is triggered when an **Ewl_Widget** (p. 785) is hidden from the **Ewl_Container** (p. 618).

6.14.3.3 `typedef void(*) Ewl_Child_Remove(Ewl_Container *c, Ewl_Widget *w, int idx)`

A typedef to shorten the definition of the `child_remove` callbacks. This callback is container specific and is triggered when an **Ewl_Widget** (p. 785) is removed from the **Ewl_Container** (p. 618).

6.14.3.4 `typedef void(*) Ewl_Child_Resize(Ewl_Container *c, Ewl_Widget *w, int size, Ewl_Orientation o)`

A typedef to shorten the definition of the `child_resize` callbacks. This callback is container specific and is triggered when an **Ewl_Widget** (p. 785) is resized in the **Ewl_Container** (p. 618).

6.14.3.5 `typedef void(*) Ewl_Child_Show(Ewl_Container *c, Ewl_Widget *w)`

A typedef to shorten the definition of the `child_show` callbacks. This callback is container specific and is triggered when an **Ewl_Widget** (p. 785) is shown to the **Ewl_Container** (p. 618).

6.14.3.6 `typedef struct Ewl_Container Ewl_Container`

This class inherits from **Ewl_Widget** (p. 785) and provides the capabilities necessary for nesting other widgets inside.

6.14.3.7 `typedef Ewl_Widget*(*) Ewl_Container_Iterator(Ewl_Container *c)`

A typedef to shorten the definition of the `child_iterator` callbacks. This callback is container specific and is usually set in the container's `init` function. This is used to pick the next child in the list of children for the container.

6.14.4 Function Documentation

6.14.4.1 `void ewl_container_add_notify_set (Ewl_Container * container, Ewl_Child_Add add)`

Set the function to be called when adding children.

Parameters:

container,: the container to change the add notifier

add,: the new add notifier for the container

Returns:

Returns no value.

Changes the add notifier function of *container* to *add*.

6.14.4.2 void ewl_container_callback_intercept (Ewl_Container * *c*, unsigned int *t*)

Receive a callback in place of children.

Parameters:

c,: the container to intercept the callback

t,: the type of callback to intercept

Returns:

Returns no value.

Marks the callbacks of type *t* that are directed to children to be triggered on the container *c*, and not propagated to the receiving child.

6.14.4.3 void ewl_container_callback_nointercept (Ewl_Container * *c*, unsigned int *t*)

Remove a callback interception from children.

Parameters:

c,: the container to remove the interception

t,: the type of callback to not intercept

Returns:

Returns no value.

Marks the callbacks of type *t* that are directed to children to be propagated to the receiving child.

6.14.4.4 void ewl_container_callback_notify (Ewl_Container * *c*, unsigned int *t*)

receive a callback of a child

Parameters:

c,: the container to notify the callback

t,: the type of callback to notify

Returns:

Returns no value.

Marks the callbacks of type *t* that are directed to children to be triggered on the container *c*, and propagated to the receiving child.

6.14.4.5 `void ewl_container_cb_configure (Ewl_Widget * w, void * ev_data, void * user_data)`

6.14.4.6 `void ewl_container_cb_disable (Ewl_Widget * w, void * ev_data, void * user_data)`

6.14.4.7 `void ewl_container_cb_enable (Ewl_Widget * w, void * ev_data, void * user_data)`

6.14.4.8 `void ewl_container_cb_obscure (Ewl_Widget * w, void * ev_data, void * user_data)`

6.14.4.9 `void ewl_container_cb_realize (Ewl_Widget * w, void * ev_data, void * user_data)`

6.14.4.10 `void ewl_container_cb_reparent (Ewl_Widget * w, void * ev_data, void * user_data)`

6.14.4.11 `void ewl_container_cb_reveal (Ewl_Widget * w, void * ev_data, void * user_data)`

6.14.4.12 `void ewl_container_cb_unrealize (Ewl_Widget * w, void * ev_data, void * user_data)`

6.14.4.13 `void ewl_container_child_add_call (Ewl_Container * c, Ewl_Widget * w)`

Triggers the `child_add` callback for the container *c*.

Parameters:

c,: the container receiving a new child widget

w,: the child widget added to the container

Returns:

Returns no value.

6.14.4.14 `void ewl_container_child_append (Ewl_Container * pc, Ewl_Widget * child)`

Add a child at the end of the container.

Parameters:

pc,: the parent container that will hold the child

child,: the child to add to the container

Returns:

Returns no value.

Attaches the child to the end of the parent containers child list.

6.14.4.15 Ewl_Widget* ewl_container_child_at_get (Ewl_Container * *widget*, int *x*, int *y*)

Find the child at the given coordinates.

Parameters:

widget,: the searched container

x,: the x coordinate to search for

y,: the y coordinate to search for

Returns:

Returns the found widget on success, NULL on failure.

The given container is searched to find any child that intersects the given coordinates.

6.14.4.16 Ewl_Widget* ewl_container_child_at_recursive_get (Ewl_Container * *widget*, int *x*, int *y*)

Find child that intersects coordinates.

Parameters:

widget,: the widget to search for intersecting child

x,: the x coordinate of the intersection point

y,: the y coordinate of the intersection point

Returns:

Returns the intersecting widget on success, NULL on failure.

6.14.4.17 int ewl_container_child_count_get (Ewl_Container * *c*)

Returns the number of child widgets in the container.

Parameters:

c,: The container to get the child count from

Returns:

Returns the number of child widgets

6.14.4.18 `int ewl_container_child_count_internal_get (Ewl_Container * c)`

Returns the number of child widgets in the container.

Parameters:

c,: The container to get the child count from

Returns:

Returns the number of child widgets

6.14.4.19 `Ewl_Widget* ewl_container_child_get (Ewl_Container * parent, int index)`

Retrieve the child at the given index in the container.

Parameters:

parent,: The container to get the child from

index,: The child index to return

Returns:

Returns the widget at the given index, or NULL if not found

6.14.4.20 `void ewl_container_child_hide_call (Ewl_Container * c, Ewl_Widget * w)`

Triggers the child_hide callback for the container *c*.

Parameters:

c,: the container receiving a new child widget

w,: the child widget hidden in the container

Returns:

Returns no value.

6.14.4.21 `int ewl_container_child_index_get (Ewl_Container * parent, Ewl_Widget * w)`

Retrieves the index of the given child in the container or -1 if not found.

Parameters:

parent,: The container to search

w,: The child to search for

Returns:

Returns the index of the child in the parent or -1 if not found

6.14.4.22 `int ewl_container_child_index_internal_get (Ewl_Container * parent,
Ewl_Widget * w)`

Retrieves the index of the given child in the container taking internal widgets into account or -1 if not found.

Parameters:

parent,: The container to search

w,: The child to search for

Returns:

Returns the index of the child in the parent including internal widgets or -1 if not found

6.14.4.23 `void ewl_container_child_insert (Ewl_Container * pc, Ewl_Widget *
child, int index)`

Add a child at an index of the container.

Parameters:

pc,: the parent container that will hold the child

child,: the child to add to the container

index,: the position in the child list to add the child (not including internal widgets)

Returns:

Returns no value.

Attaches the child to the *index* position of the parent containers child list.

6.14.4.24 `void ewl_container_child_insert_internal (Ewl_Container * pc,
Ewl_Widget * child, int index)`

Add a child at an index of the container.

Parameters:

pc,: the parent container that will hold the child

child,: the child to add to the container

index,: the position in the child list to add the child

Returns:

Returns no value.

Attaches the child to the *index* position of the parent containers child list.

**6.14.4.25 Ewl_Widget* ewl_container_child_internal_get (Ewl_Container *
parent, int *index*)**

Retrieves the child at the given index in the container taking internal widgets into account.

Parameters:

parent,: The container to get the child from
index,: The child index to return

Returns:

Returns the widget at the given index including internal widgets, or NULL if not found

6.14.4.26 void ewl_container_child_iterate_begin (Ewl_Container * *c*)

Prepare the container to iterate over it's children.

Parameters:

c,: the container to begin iterating over it's children

Returns:

Returns no value.

Notifies the container that it will need to start from the beginning of it's child list. Do not call this if you want to begin iteration where it was last left off, but be warned it's possible this won't be where you expect.

**6.14.4.27 void ewl_container_child_iterator_set (Ewl_Container * *c*,
Ewl_Container_Iterator *i*)**

Changes the iterator function for a container.

Parameters:

c,: the container to change child iterator functions
i,: the iterator function for the container

Returns:

Returns no value.

Do not use this unless you know are writing a custom container of your own.

6.14.4.28 Ewl_Widget* ewl_container_child_next (Ewl_Container * *c*)

Retrieve the next eligible child in a container.

Parameters:

c,: the container to retrieve the next usable child

Returns:

Returns the next valid child on success, NULL on failure.

6.14.4.29 `void ewl_container_child_prepend (Ewl_Container * pc, Ewl_Widget * child)`

Add a child at the beginning of the container.

Parameters:

pc,: the parent container that will hold the child

child,: the child to add to the container

Returns:

Returns no value.

Attaches the child to the start of the parent containers child list.

6.14.4.30 `void ewl_container_child_remove (Ewl_Container * pc, Ewl_Widget * child)`

Remove the specified child from the container.

Parameters:

pc,: the container to search for the child to remove

child,: the child to remove from the container

Returns:

Returns no value.

Removes the specified child from the container without destroying the child.

6.14.4.31 `void ewl_container_child_remove_call (Ewl_Container * c, Ewl_Widget * w, int idx)`

Triggers the child_remove callback for the container *c*.

Parameters:

c,: the container removing a child widget

w,: the child widget removed from the container

idx,: The index that the child was in when it was removed

Returns:

Returns no value.

6.14.4.32 `void ewl_container_child_resize (Ewl_Widget * w, int size, Ewl_Orientation o)`

Notify a container of a child pref size change.

Parameters:

w,: the child widget that has had it's preferred size changed
size,: the amount of change in size
o,: the orientation of the size change

Returns:

Returns no value.

6.14.4.33 `void ewl_container_child_show_call (Ewl_Container * c, Ewl_Widget * w)`

Triggers the `child_show` callback for the container *c*.

Parameters:

c,: the container receiving a new child widget
w,: the child widget shown in the container

Returns:

Returns no value.

6.14.4.34 `void ewl_container_destroy (Ewl_Container * c)`

Destroy all the sub-children of the container.

Parameters:

c,: the container to destroy children

Returns:

Returns no value.

6.14.4.35 `Ewl_Container* ewl_container_end_redirect_get (Ewl_Container * c)`

Searches for the last redirected container of the container.

Parameters:

c,: the container to retrieve it's redirection end container

Returns:

Returns the container children are placed in, NULL if none.

**6.14.4.36 void ewl_container_hide_notify_set (Ewl_Container * *container*,
Ewl_Child_Hide *hide*)**

Set the function to be called when hideing children.

Parameters:

container,: the container to change the hide notifier
hide,: the new show notifier for the container

Returns:

Returns no value.

Changes the hide notifier function of *container* to *hide*.

6.14.4.37 int ewl_container_init (Ewl_Container * *c*)

Initialize a containers default fields and callbacks.

Parameters:

c,: the container to initialize

Returns:

Returns TRUE on success, otherwise FALSE.

Initializes the default values of the container, this also sets up the widget fields of the container.

**6.14.4.38 void ewl_container_largest_prefer (Ewl_Container * *c*,
Ewl_Orientation *o*)**

Set preferred size to widest child.

Parameters:

c,: the container to change preferred size
o,: the orientation/direction to prefer the largest widget

Returns:

Returns no value.

This function can be used by any container which wishes to set it's preferred width to that of it's widest child.

6.14.4.39 Ewl_Container* ewl_container_redirect_get (Ewl_Container * *c*)

Retrieves for the redirected container of the container.

Parameters:

c,: the container to retrieve it's redirection container

Returns:

Returns the container children are placed in, NULL if none.

6.14.4.40 `void ewl_container_redirect_set (Ewl_Container * c, Ewl_Container * rc)`

Changes the redirected container of the container.

Parameters:

c,: the container to change it's redirection container

rc,: the new container to redirect children to

Returns:

Returns no value.

6.14.4.41 `void ewl_container_remove_notify_set (Ewl_Container * container, Ewl_Child_Remove remove)`

Set the function to be called when removing children.

Parameters:

container,: the container to change the add notifier

remove,: the new remove notifier for the container

Returns:

Returns no value.

Changes the remove notifier function of *container* to *remove*.

6.14.4.42 `void ewl_container_reset (Ewl_Container * c)`

Destroy all children of the container.

Parameters:

c,: the container to reset

Returns:

Returns no value.

Destroys all the children of the container but not the container itself or internally used widgets.

6.14.4.43 `void ewl_container_resize_notify_set (Ewl_Container * container, Ewl_Child_Resize resize)`

Set function to be called when children resize.

Parameters:

container,: the container to change the resize notifier

resize,: the new resizenotifier for the container

Returns:

Returns no value.

Changes the resize notifier function of *container* to *resize*.

6.14.4.44 `void ewl_container_show_notify_set (Ewl_Container * container,
Ewl_Child_Show show)`

Set the function to be called when showing children.

Parameters:

container,: the container to change the show notifier

show,: the new show notifier for the container

Returns:

Returns no value.

Changes the show notifier function of *container* to *show*.

6.14.4.45 `void ewl_container_sum_prefer (Ewl_Container * c, Ewl_Orientation
o)`

Set the sum orientation of the container.

Parameters:

c,: the container to use the child size sum in a specified direction

o,: the orientation direction of the sum to use

Returns:

Returns no value

6.15 Ewl_Context_Menu: A context menu

6.15.1 Detailed Description

Remarks:

Inherits from **Ewl_Popup** (p. 721).

Data Structures

- struct **Ewl_Context_Menu**

*Inherits from **Ewl_Popup** (p. 721) and extends to provide a context menu.*

Defines

- `#define EWL_CONTEXT_MENU(context_menu) ((Ewl_Context_Menu *) context_menu)`
- `#define EWL_CONTEXT_MENU_IS(w)`
- `#define EWL_CONTEXT_MENU_TYPE "context_menu"`

Typedefs

- `typedef Ewl_Context_Menu Ewl_Context_Menu`

Functions

- `void ewl_context_menu_attach (Ewl_Context_Menu *cm, Ewl_Widget *w)`
- `void ewl_context_menu_cb_attach_mouse_down (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_context_menu_cb_child_add (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_context_menu_cb_child_clicked (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_context_menu_cb_child_mouse_in (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_context_menu_cb_child_remove (Ewl_Container *c, Ewl_Widget *w, int idx)`
- `void ewl_context_menu_cb_focus_in (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_context_menu_cb_hide (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_context_menu_cb_mouse_down (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_context_menu_cb_mouse_move (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_context_menu_container_set (Ewl_Context_Menu *cm, Ewl_Container *c)`
set a custom container for the context menu
- `void ewl_context_menu_detach (Ewl_Context_Menu *cm, Ewl_Widget *w)`

- `int ewl_context_menu_init (Ewl_Context_Menu *cm)`

Initialize the context_menu and inherited fields.

- `Ewl_Widget * ewl_context_menu_new (void)`

Allocate a new context_menu widget.

6.15.2 Define Documentation

- ### 6.15.2.1 `#define EWL_CONTEXT_MENU(context_menu) ((Ewl_Context_Menu *) context_menu)`

Typecast a pointer to an `Ewl_Context_Menu` (p. 620) pointer.

- ### 6.15.2.2 `#define EWL_CONTEXT_MENU_IS(w)`

Value:

```
(ewl_widget_type_is(EWL_WIDGET(w), \
                                EWL_CONTEXT_MENU_TYPE))
```

Returns TRUE if the widget is an `Ewl_Context_Menu` (p. 620), FALSE otherwise

- ### 6.15.2.3 `#define EWL_CONTEXT_MENU_TYPE "context_menu"`

The type name for the `Ewl_Context_Menu` (p. 620) widget

6.15.3 Typedef Documentation

- ### 6.15.3.1 `typedef struct Ewl_Context_Menu Ewl_Context_Menu`

`Ewl_Context_Menu` (p. 620)

6.15.4 Function Documentation

- ### 6.15.4.1 `void ewl_context_menu_attach (Ewl_Context_Menu * cm, Ewl_Widget * w)`

Parameters:

cm,: the context menu to attach

w,: a widget that gets the context menu attached

This function attaches a context menu to a widget. You can attach one context menu to more than one widget but all widgets have to share the same embed. i.e. they must be in the same window or embed.

- 6.15.4.2 `void ewl_context_menu_cb_attach_mouse_down (Ewl_Widget * w,
void * ev_data, void * user_data)`
- 6.15.4.3 `void ewl_context_menu_cb_child_add (Ewl_Container * c,
Ewl_Widget * w)`
- 6.15.4.4 `void ewl_context_menu_cb_child_clicked (Ewl_Widget * w, void *
ev_data, void * user_data)`
- 6.15.4.5 `void ewl_context_menu_cb_child_mouse_in (Ewl_Widget * w, void *
ev_data, void * user_data)`
- 6.15.4.6 `void ewl_context_menu_cb_child_remove (Ewl_Container * c,
Ewl_Widget * w, int idx)`
- 6.15.4.7 `void ewl_context_menu_cb_focus_in (Ewl_Widget * w, void * ev_data,
void * user_data)`
- 6.15.4.8 `void ewl_context_menu_cb_hide (Ewl_Widget * w, void * ev_data,
void * user_data)`
- 6.15.4.9 `void ewl_context_menu_cb_mouse_down (Ewl_Widget * w, void *
ev_data, void * user_data)`
- 6.15.4.10 `void ewl_context_menu_cb_mouse_move (Ewl_Widget * w, void *
ev_data, void * user_data)`
- 6.15.4.11 `void ewl_context_menu_container_set (Ewl_Context_Menu * cm,
Ewl_Container * c)`

set a custom container for the context menu

Parameters:

cm,: the context menu to set the container

c,: the container to use inside of the context menu, if *c* is NULL then this function will use a vbox instead

This function give you the ability to set a custom container as the internal used container of the context menu. It is not necessary to use this function, because the context menu uses a vbox as default, if you do not add an own one. If you override a previous added container, this container will be destroyed, but the children will be moved into the new container.

- 6.15.4.12 `void ewl_context_menu_detach (Ewl_Context_Menu * cm,
Ewl_Widget * w)`

Parameters:

cm,: the context menu to detach

w,: a widget that has *cm* attached

This function detachs an earlier attached context menu to a widget

6.15.4.13 int ewl_context_menu_init (Ewl_Context_Menu * *cm*)

Initialize the context_menu and inherited fields.

Parameters:

cm,: the context_menu to initialize

Returns:

Returns TRUE on success, FALSE on failure.

Clears the contents of the context_menu and stores the default values.

6.15.4.14 Ewl_Widget* ewl_context_menu_new (void)

Allocate a new context_menu widget.

Returns:

Returns pointer to new context_menu widget on success, NULL on failure.

6.16 Ewl_Cursor: Container to create custom mouse cursors

6.16.1 Detailed Description

Defines the **Ewl_Cursor** (p. 622) class used for generating ARGB cursor data.

Remarks:

Inherits from **Ewl_Window** (p. 788).

Data Structures

- struct **Ewl_Cursor**

*Inherits from **Ewl_Window** (p. 788) to allow packing widgets as cursor objects.*

Defines

- `#define EWL_CURSOR(cursor) ((Ewl_Cursor *) cursor)`
- `#define EWL_CURSOR_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_CURSOR_TYPE))`
- `#define EWL_CURSOR_TYPE "cursor"`

Typedefs

- `typedef Ewl_Cursor Ewl_Cursor`

Functions

- `int ewl_cursor_init (Ewl_Cursor *c)`
*Initializes an **Ewl_Cursor** (p. 622) widget.*
- `Ewl_Widget * ewl_cursor_new (void)`
*Creates and initializes a new **Ewl_Cursor** (p. 622) widget.*

6.16.2 Define Documentation

6.16.2.1 `#define EWL_CURSOR(cursor) ((Ewl_Cursor *) cursor)`

Typecast a pointer to an **Ewl_Separator** (p. 740) pointer.

6.16.2.2 `#define EWL_CURSOR_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_CURSOR_TYPE))`

Returns TRUE if the widget is an **Ewl_Cursor** (p. 622), FALSE otherwise

6.16.2.3 #define EWL_CURSOR_TYPE "cursor"

The type name for the **Ewl_Cursor** (p. 622) widget

6.16.3 Typedef Documentation

6.16.3.1 typedef struct Ewl_Cursor Ewl_Cursor

Ewl_Cursor (p. 622) provides a container that has a labeled cursor.

6.16.4 Function Documentation

6.16.4.1 int ewl_cursor_init (Ewl_Cursor * *cursor*)

Initializes an **Ewl_Cursor** (p. 622) widget.

Parameters:

cursor,: The **Ewl_Cursor** (p. 622) to initialize

Returns:

Returns TRUE on success, FALSE on failure

6.16.4.2 Ewl_Widget* ewl_cursor_new (void)

Creates and initializes a new **Ewl_Cursor** (p. 622) widget.

Returns:

Returns a new **Ewl_Cursor** (p. 622) on success, or NULL on failure

6.17 Ewl_Datepicker: The EWL Datepicker widget

6.17.1 Detailed Description

Defines the **Ewl_Datepicker** (p. 624) class,.

Remarks:

Inherits from **Ewl_Text** (p. 757).

Data Structures

- struct **Ewl_Datepicker**

*Inherits from **Ewl_Text** (p. 757) and extends to provide a date picker.*

Defines

- #define **EWL_DATEPICKER**(datepicker) ((**Ewl_Datepicker** *) datepicker)
- #define **EWL_DATEPICKER_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_DATEPICKER_TYPE))
- #define **EWL_DATEPICKER_TYPE** "datepicker"

Typedefs

- typedef **Ewl_Datepicker** **Ewl_Datepicker**

Functions

- void **ewl_datepicker_cb_destroy** (**Ewl_Widget** *w, void *ev, void *user_data)
- void **ewl_datepicker_cb_value_changed** (**Ewl_Widget** *w, void *ev, void *user_data)
- void **ewl_datepicker_cb_window_mouse_down** (**Ewl_Widget** *w, void *ev, void *user_data)
- int **ewl_datepicker_init** (**Ewl_Datepicker** *datepicker)
Initialize the date picker to default values.
- **Ewl_Widget** * **ewl_datepicker_new** (void)
*Creates a new **Ewl_Datepicker** (p. 624).*

6.17.2 Define Documentation

- #### 6.17.2.1 #define **EWL_DATEPICKER**(datepicker) ((**Ewl_Datepicker** *) datepicker)

Typecast a pointer to an **Ewl_Datepicker** (p. 624) pointer

6.17.2.2 `#define EWL_DATEPICKER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_DATEPICKER_TYPE))`

Returns TRUE if the widget is an **Ewl_Datepicker** (p. 624), FALSE otherwise

6.17.2.3 `#define EWL_DATEPICKER_TYPE "datepicker"`

The type name for the **Ewl_Datepicker** (p. 624) widget

6.17.3 Typedef Documentation

6.17.3.1 `typedef struct Ewl_Datepicker Ewl_Datepicker`

The **Ewl_Datepicker** (p. 624)

6.17.4 Function Documentation

6.17.4.1 `void ewl_datepicker_cb_destroy (Ewl_Widget * w, void * ev, void * user_data)`

6.17.4.2 `void ewl_datepicker_cb_value_changed (Ewl_Widget * w, void * ev, void * user_data)`

6.17.4.3 `void ewl_datepicker_cb_window_mouse_down (Ewl_Widget * w, void * ev, void * user_data)`

6.17.4.4 `int ewl_datepicker_init (Ewl_Datepicker * dp)`

Initialize the date picker to default values.

Parameters:

dp,: The **Ewl_Datepicker** (p. 624) to initialize

Returns:

Returns TRUE on success or FALSE on failure

6.17.4.5 `Ewl_Widget* ewl_datepicker_new (void)`

Creates a new **Ewl_Datepicker** (p. 624).

Returns:

Returns NULL on failure, a new **Ewl_Datepicker** (p. 624) on success

6.18 Ewl_Dialog: A dialog window

6.18.1 Detailed Description

Defines the **Ewl_Dialog** (p. 626) class which extends the **Ewl_Window** (p. 788) class.

Remarks:

Inherits from **Ewl_Window** (p. 788).

Data Structures

- struct **Ewl_Dialog**
*Inherits from **Ewl_Window** (p. 788) and extends to provide a dialog widget.*

Defines

- #define **EWL_DIALOG**(dialog) ((**Ewl_Dialog** *) dialog)
- #define **EWL_DIALOG_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_DIALOG_TYPE))
- #define **EWL_DIALOG_TYPE** "dialog"

Typedefs

- typedef **Ewl_Dialog** **Ewl_Dialog**

Functions

- **Ewl_Position** ewl_dialog_action_position_get (**Ewl_Dialog** *dialog)
Checks the action area position for a dialog.
- void ewl_dialog_action_position_set (**Ewl_Dialog** *d, **Ewl_Position** pos)
Changes the action area position for a dialog.
- **Ewl_Position** ewl_dialog_active_area_get (**Ewl_Dialog** *d)
Retrieve the location of the dialog widgets will be packed into.
- void ewl_dialog_active_area_set (**Ewl_Dialog** *d, **Ewl_Position** pos)
Sets the section of the dialog that will have widgets packed into it.
- unsigned int ewl_dialog_has_separator_get (**Ewl_Dialog** *dialog)
Checks if dialog has a separator or not.
- void ewl_dialog_has_separator_set (**Ewl_Dialog** *dialog, unsigned int has_sep)
Sets the separator of dialog.
- int ewl_dialog_init (**Ewl_Dialog** *dialog)
Initialize an internal dialog to starting values.

- **Ewl_Widget * ewl_dialog_new** (void)

Create a new internal dialog.

6.18.2 Define Documentation

6.18.2.1 #define EWL_DIALOG(dialog) ((Ewl_Dialog *) dialog)

Typecasts a pointer to an **Ewl_Dialog** (p. 626) pointer.

6.18.2.2 #define EWL_DIALOG_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_DIALOG_TYPE))

Returns TRUE if the widget is an **Ewl_Dialog** (p. 626), FALSE otherwise

6.18.2.3 #define EWL_DIALOG_TYPE "dialog"

The type name for the **Ewl_Dialog** (p. 626) widget

6.18.3 Typedef Documentation

6.18.3.1 typedef struct Ewl_Dialog Ewl_Dialog

The dialog structure is a window with two areas: a box (vbox) to hold messages and another box (action_area) to put buttons (usually). These boxes can be separated by a line.

6.18.4 Function Documentation

6.18.4.1 Ewl_Position ewl_dialog_action_position_get (Ewl_Dialog * d)

Checks the action area position for a dialog.

Parameters:

d,: dialog to check action area position

Returns:

Returns the current action area position.

6.18.4.2 void ewl_dialog_action_position_set (Ewl_Dialog * d, Ewl_Position pos)

Changes the action area position for a dialog.

Parameters:

d,: dialog to change action area position

pos,: the new position for the new action area

Returns:

Returns no value.

6.18.4.3 Ewl_Position ewl_dialog_active_area_get (Ewl_Dialog * *d*)

Retrieve the location of the dialog widgets will be packed into.

Parameters:

d,: The **Ewl_Dialog** (p. 626) to get the active area from

Returns:

Returns the active portion of the dialog

6.18.4.4 void ewl_dialog_active_area_set (Ewl_Dialog * *d*, Ewl_Position *pos*)

Sets the section of the dialog that will have widgets packed into it.

Parameters:

d,: The **Ewl_Dialog** (p. 626) to work with

pos,: The area of the dialog to set active

Returns:

Returns no value

6.18.4.5 unsigned int ewl_dialog_has_separator_get (Ewl_Dialog * *dialog*)

Checks if *dialog* has a separator or not.

Parameters:

dialog,: the dialog.

Returns:

Returns TRUE if *dialog* has a separator.

6.18.4.6 void ewl_dialog_has_separator_set (Ewl_Dialog * *dialog*, unsigned int *has_sep*)

Sets the separator of *dialog*.

Parameters:

dialog,: the dialog.

has_sep,: TRUE to draw the separator, FALSE to hide it.

Returns:

Returns no value.

6.18.4.7 int ewl_dialog_init (Ewl_Dialog * *dialog*)

Initialize an internal dialog to starting values.

Parameters:

dialog,: the dialog to initialize.

Returns:

Return TRUE on success, FALSE otherwise.

6.18.4.8 Ewl_Widget* ewl_dialog_new (void)

Create a new internal dialog.

Returns:

Returns a pointer to a new dialog on success, NULL on failure.

6.19 Ewl_Dnd: The files containing DND functions

6.19.1 Detailed Description

6.19.2 Tutorial

In order to make DND handling easier for the end programmer, we've added a simplification API to EWL. It currently supports receiving drops and some limited dragging of widgets. Getting started is fairly simple, even for some rather complicated widgets. As an example, I'll run through the DND support added to the entry widget.

The first step to setup DND support on a widget is to decide which MIME types are allowed and understood by the widget. The entry displays text, so accepting the type "text/plain" is a safe choice. A NULL terminated list of type strings is passed to the `ewl_dnd_accepted_types_set`, which enables DND responses for the widget and helps to negotiate whether a drop will be accepted at a given position.

```
const char *text_types[] = { "text/plain", NULL };

ewl_dnd_accepted_types_set(EWL_WIDGET(e), text_types);
```

One key feature for DND support in the entry widget was to allow dragging text to arbitrary positions within the visible text area. This is accomplished by registering a callback for the `EWL_CALLBACK_DND_POSITION` event on the entry widget.

```
ewl_callback_append(w, EWL_CALLBACK_DND_POSITION, ewl_entry_cb_dnd_position, NULL);
```

When the mouse moves during a DND event over the specified entry `w` the `ewl_entry_cb_dnd_position` function will be called. This function prototype looks like all other EWL callback prototypes:

```
void ewl_entry_cb_dnd_position(Ewl_Widget *w, void *ev, void *data);
```

In this case, the `void *ev` parameter points to a **Ewl_Event_Dnd_Position** (p.643) struct, which contains more detailed information about the event. We can use the coordinates from the event to position the cursor within our entry to receive the dropped data. Since the entry widget inherits from the text widget, the text calls are used directly on the widget to alter the entry contents. The code to accomplish this is rather small when the extra debugging information is removed:

```
void
ewl_entry_cb_dnd_position(Ewl_Widget *w, void *ev, void *data)
{
    Ewl_Event_Dnd_Position *event;
    Ewl_Text *txt;

    event = ev;
    txt = EWL_TEXT(w);

    if (EWL_ENTRY(w)->editable && !DISABLED(w)) {
        ewl_widget_focus_send(w);
        ewl_text_cursor_position_set(txt, ewl_text_coord_index_map(txt, event->x, event->y));
    }
}
```

Once the cursor has been positioned, the only event we care about is receiving the data from the drop. This is accomplished by using the `EWL_CALLBACK_DND_DATA` callback which should also be placed on the entry widget.


```
ewl_callback_append(w, EWL_CALLBACK_DND_DATA, ewl_entry_cb_dnd_data, NULL);
```

The function prototype for `ewl_entry_cb_dnd_data` is identical to `ewl_entry_cb_dnd_position`, but the void `*ev` parameter is of type `Ewl_Event_Dnd_Data`. Since we only registered to receive plain text data dropped on the entry, we can insert the event data directly into the entry at the current cursor position.

```
void
ewl_entry_cb_dnd_data(Ewl_Widget *w, void *ev, void *data)
{
    Ewl_Event_Dnd_Data *event;
    Ewl_Text *txt;

    event = ev;
    txt = EWL_TEXT(w);

    if (EWL_ENTRY(w)->editable && !DISABLED(w)) {
        ewl_text_text_insert(txt, event->data,
                            ewl_text_cursor_position_get(txt));
    }
}
```

Considering the complicated nature of the Xdnd protocol, we are able to accomplish a considerable amount of work in very few lines of code. While some flexibility is sacrificed to achieve this, almost all of the protocol events are available for widgets to override as they please.

Check back for followup information to handle drag events on widgets.

Functions

- **int ewl_dnd_accepted_types_contains** (**Ewl_Widget** *w, **char** *type)
: *Verifies the specified widget accepts the given mimetype*
- **const char ** ewl_dnd_accepted_types_get** (**Ewl_Widget** *w)
: *Gets the mimetypes the designated widget can accept for DND*
- **void ewl_dnd_accepted_types_set** (**Ewl_Widget** *w, **const char **types**)
: *Sets the mimetypes the designated widget can accept for DND*
- **void ewl_dnd_disable** (**void**)
: *Disables DND.*
- **void ewl_dnd_drag_drop** (**Ewl_Widget** *w)
: *Tells the widget to start dragging.*
- **void ewl_dnd_drag_start** (**Ewl_Widget** *w)
: *Tells the widget to start dragging.*
- **void ewl_dnd_drag_widget_clear** (**void**)
: *Clears the current DND widget.*
- **Ewl_Widget * ewl_dnd_drag_widget_get** (**void**)
: *Retrieves the current DND widget.*

- `void ewl_dnd_enable (void)`
Enables DND.
- `int ewl_dnd_init (void)`
- `int ewl_dnd_provided_types_contains (Ewl_Widget *w, char *type)`
: Verifies the specified widget provides the given mimetype
- `char ** ewl_dnd_provided_types_get (Ewl_Widget *w)`
: Gets the mimetypes the designated widget can provide for DND
- `void ewl_dnd_provided_types_set (Ewl_Widget *w, const char **types)`
: Sets the mimetypes the designated widget can provide for DND
- `void ewl_dnd_shutdown (void)`
- `int ewl_dnd_status_get (void)`
Retrieves the current DND status.

Variables

- `int EWL_CALLBACK_DND_DATA_RECEIVED`
- `int EWL_CALLBACK_DND_DATA_REQUEST`
- `int EWL_CALLBACK_DND_DROP`
- `int EWL_CALLBACK_DND_ENTER`
- `int EWL_CALLBACK_DND_LEAVE`
- `int EWL_CALLBACK_DND_POSITION`

6.19.3 Function Documentation

6.19.3.1 `int ewl_dnd_accepted_types_contains (Ewl_Widget * w, char * type)`

: Verifies the specified widget accepts the given mimetype

Parameters:

- w*,: The widget to test for an accepted type
type,: The mimetype to test for acceptance on a specific widget

Returns:

Returns TRUE if the widget accepts the given type, FALSE otherwise

6.19.3.2 `const char** ewl_dnd_accepted_types_get (Ewl_Widget * w)`

: Gets the mimetypes the designated widget can accept for DND

Parameters:

- w*,: The widget to retrieve accepted types

Returns:

Returns a NULL terminated array of mimetypes widget accepts for DND

**6.19.3.3 void ewl_dnd_accepted_types_set (Ewl_Widget * *w*, const char **
types)**

: Sets the mimetypes the designated widget can accept for DND

Parameters:

w,: The widget to set accepted types

types,: A NULL terminated array of mimetypes widget accepts for DND

Returns:

Returns no value

6.19.3.4 void ewl_dnd_disable (void)

Disables DND.

Returns:

Returns no value

6.19.3.5 void ewl_dnd_drag_drop (Ewl_Widget * *w*)

Tells the widget to start dragging.

Parameters:

w,: The widget to start dragging

Returns:

Returns no value

6.19.3.6 void ewl_dnd_drag_start (Ewl_Widget * *w*)

Tells the widget to start dragging.

Parameters:

w,: The widget to start dragging

Returns:

Returns no value

6.19.3.7 void ewl_dnd_drag_widget_clear (void)

Clears the current DND widget.

Returns:

Returns no value.

6.19.3.8 Ewl_Widget* ewl_dnd_drag_widget_get (void)

Retrieves the current DND widget.

Returns:

Returns the current DND widget

6.19.3.9 void ewl_dnd_enable (void)

Enables DND.

Returns:

Returns no value

6.19.3.10 int ewl_dnd_init (void)**6.19.3.11 int ewl_dnd_provided_types_contains (Ewl_Widget * *w*, char * *type*)**

: Verifies the specified widget provides the given mimetype

Parameters:

w,: The widget to test for an provided type

type,: The mimetype to test for provideance on a specific widget

Returns:

Returns TRUE if the types contains the given type, FALSE otherwise

6.19.3.12 char ewl_dnd_provided_types_get (Ewl_Widget * *w*)**

: Gets the mimetypes the designated widget can provide for DND

Parameters:

w,: The widget to retrieve provided types

Returns:

Returns a NULL terminated array of mimetypes widget provides for DND

6.19.3.13 void ewl_dnd_provided_types_set (Ewl_Widget * *w*, const char ** *types*)

: Sets the mimetypes the designated widget can provide for DND

Parameters:

w,: The widget to set provided types

types,: A NULL terminated array of mimetypes widget provides for DND

Returns:

Returns no value

6.19.3.14 void ewl_dnd_shutdown (void)

6.19.3.15 int ewl_dnd_status_get (void)

Retrieves the current DND status.

Returns:

Returns the current DND status

6.19.4 Variable Documentation

6.19.4.1 int EWL_CALLBACK_DND_DATA_RECEIVED

Data received event

6.19.4.2 int EWL_CALLBACK_DND_DATA_REQUEST

Data request event

6.19.4.3 int EWL_CALLBACK_DND_DROP

Drop event

6.19.4.4 int EWL_CALLBACK_DND_ENTER

On enter of a widget

6.19.4.5 int EWL_CALLBACK_DND_LEAVE

On exit of a widget

6.19.4.6 int EWL_CALLBACK_DND_POSITION

A DND position event

6.20 Ewl_Embed: A Container for Displaying on an canvas

6.20.1 Detailed Description

Defines the **Ewl_Embed** (p. 629) class to provide EWL with the ability to work with a canvas.

Remarks:

Inherits from **Ewl_Overlay** (p. 712).

6.20.2 Tutorial

The embed allows for overriding the engine used for rendering and display so that an application can have multiple target engines. This is most commonly used through the window layer as the embed itself does not create windows in the display system. The reason it is not implemented at the window level is that the embed will need the correct engine information in order to interact with menus, tooltips and similar widgets.

```
Ewl_Widget *window;
window = ewl_window_new();
ewl_embed_engine_name_set(EWL_EMBED(window), "evas_buffer");
ewl_widget_show(window);
```

The code above demonstrates how to setup an embed to render to an ARGB data buffer. When the window is realized it will create a data buffer and store that as the window information on the embed.

Data Structures

- struct **Ewl_Embed**

*Inherits from **Ewl_Overlay** (p. 712) that acts as a top level widget for interacting with the canvas.*

Defines

- #define **EWL_EMBED**(widget) ((**Ewl_Embed** *) widget)
*Typecast a pointer to an **Ewl_Embed** (p. 629) pointer.*
- #define **EWL_EMBED_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_EMBED_TYPE))
- #define **EWL_EMBED_TYPE** "embed"
- #define **EWL_EMBED_WINDOW**(window) ((**Ewl_Embed_Window** *)window)
Typecast a pointer to an canvas window.

Typedefs

- typedef **Ewl_Embed** **Ewl_Embed**
- typedef void * **Ewl_Embed_Window**

Functions

- **Ewl_Embed * ewl_embed_active_embed_get** (void)
Returns the currently active embed widget.
- **void ewl_embed_active_set** (Ewl_Embed *embed, unsigned int act)
Sets if the given embed embed is active or not.
- **void ewl_embed_cache_cleanup** (Ewl_Embed *emb)
This will destroy the object cache associated with the embed object.
- **void * ewl_embed_canvas_set** (Ewl_Embed *emb, void *canvas, Ewl_Embed_Window *canvas_window)
Change the canvas used by the embedded container.
- **Ewl_Embed * ewl_embed_canvas_window_find** (Ewl_Embed_Window *win)
Find an ewl embed by its evas window.
- **void ewl_embed_cb_configure** (Ewl_Widget *w, void *ev_data, void *user_data)
- **void ewl_embed_cb_destroy** (Ewl_Widget *w, void *ev_data, void *user_data)
- **void ewl_embed_cb_focus_out** (Ewl_Widget *w, void *ev_data, void *user_data)
- **void ewl_embed_cb_realize** (Ewl_Widget *w, void *ev_data, void *user_data)
- **void ewl_embed_cb_unrealize** (Ewl_Widget *w, void *ev_data, void *user_data)
- **void ewl_embed_coord_to_screen** (Ewl_Embed *e, int xx, int yy, int *x, int *y)
Maps coordinates from the Evas to screen coordinates.
- **void ewl_embed_desktop_size_get** (Ewl_Embed *e, int *w, int *y)
Retrieve the size of the root window.
- **void ewl_embed_dnd_aware_remove** (Ewl_Embed *embed)
Cancels an embed as being DND aware.
- **void ewl_embed_dnd_aware_set** (Ewl_Embed *embed)
Set an embed as being DND aware.
- **void ewl_embed_dnd_data_received_feed** (Ewl_Embed *embed, char *type, void *data, unsigned int len, unsigned int format)
Sends the event for selection data received into an embed.
- **void ewl_embed_dnd_data_request_feed** (Ewl_Embed *embed, void *handle, char *type)
Sends the request event for selection data received into an embed.
- **const char * ewl_embed_dnd_drop_feed** (Ewl_Embed *embed, int x, int y, int internal)
Sends the event for a DND drop into an embed.
- **const char * ewl_embed_dnd_position_feed** (Ewl_Embed *embed, int x, int y, int *, int *, int *, int *)

Sends the event for a DND position into an embed.

- `const char * ewl_embed_engine_name_get (Ewl_Embed *embed)`
Get the current engine on an embed.
- `int ewl_embed_engine_name_set (Ewl_Embed *embed, const char *engine)`
Changes the current engine on an embed to the specified engine.
- `int ewl_embed_focus_get (Ewl_Embed *embed)`
Retrieve the boolean value that indicates if the embed takes focus.
- `void ewl_embed_focus_set (Ewl_Embed *embed, int f)`
Sets the boolean flag in the embed to determine if it takes focus.
- `Ewl_Widget * ewl_embed_focused_widget_get (Ewl_Embed *e)`
Get the current focused widget in the embed.
- `void ewl_embed_focused_widget_set (Ewl_Embed *e, Ewl_Widget *w)`
Set the current focused widget in the embed.
- `void ewl_embed_font_path_add (char *path)`
Add a font path to all embeds after realized.
- `void ewl_embed_freeze (Ewl_Embed *e)`
Freeze redraws and events on the specified embed.
- `void ewl_embed_info_widgets_cleanup (Ewl_Embed *e, Ewl_Widget *w)`
This will check to see if the given widget is one of the last selected, clicked, mouse_in or the drag-n-drop widget and if so, set them to NULL.
- `int ewl_embed_init (Ewl_Embed *emb)`
initialize a embed to default values and callbacks
- `void ewl_embed_key_down_feed (Ewl_Embed *embed, const char *keyname, unsigned int modifiers)`
Sends the event for a key press into an embed.
- `void ewl_embed_key_up_feed (Ewl_Embed *embed, const char *keyname, unsigned int modifiers)`
Sends the event for a key release into an embed.
- `void ewl_embed_mouse_cursor_set (Ewl_Widget *w)`
This will retrieve the attached cursor off of the given widget and display it. If there is no widget on the widget it will use the default cursor.
- `void ewl_embed_mouse_down_feed (Ewl_Embed *embed, int b, int clicks, int x, int y, unsigned int modifiers)`
Sends the event for a mouse button press into an embed.
- `void ewl_embed_mouse_move_feed (Ewl_Embed *embed, int x, int y, unsigned int modifiers)`

Sends the event for a mouse button release into an embed.

- void **ewl_embed_mouse_out_feed** (Ewl_Embed *embed, int x, int y, unsigned int modifiers)

Sends a mouse out event to the last focused widget.

- void **ewl_embed_mouse_up_feed** (Ewl_Embed *embed, int b, int x, int y, unsigned int modifiers)

Sends the event for a mouse button release into an embed.

- void **ewl_embed_mouse_wheel_feed** (Ewl_Embed *embed, int x, int y, int z, int dir, unsigned int mods)

Sends a mouse out event to the last focused widget.

- **Ewl_Widget *ewl_embed_new** (void)

Allocate and initialize a new embed.

- void **ewl_embed_object_cache** (Ewl_Embed *e, void *obj)

Caches the specified object for later reuse.

- void * **ewl_embed_object_request** (Ewl_Embed *e, char *type)

Retrieves an object of type type from the embed cache, or NULL if none found.

- void **ewl_embed_position_get** (Ewl_Embed *e, int *x, int *y)

Retrieve the position of the embed.

- void **ewl_embed_shutdown** (void)

- void **ewl_embed_tab_order_append** (Ewl_Embed *e, Ewl_Widget *w)

Moves the widget w to the end of the tab order.

- void **ewl_embed_tab_order_insert** (Ewl_Embed *e, Ewl_Widget *w, unsigned int idx)

Moves the given widget w to the position idx.

- void **ewl_embed_tab_order_insert_after** (Ewl_Embed *e, Ewl_Widget *w, Ewl_Widget *after)

Inserts the w widget into the tab order after the after widget.

- void **ewl_embed_tab_order_insert_before** (Ewl_Embed *e, Ewl_Widget *w, Ewl_Widget *before)

Inserts the w widget into the tab order before the before widget.

- void **ewl_embed_tab_order_next** (Ewl_Embed *e)

Changes focus to the next widget in the circular tab order list.

- void **ewl_embed_tab_order_prepend** (Ewl_Embed *e, Ewl_Widget *w)

Moves the widget w to the front of the tab order list.

- void **ewl_embed_tab_order_previous** (Ewl_Embed *e)

Changes focus to the next widget in the circular tab order list.

- `void ewl_embed_tab_order_remove (Ewl_Embed *e, Ewl_Widget *w)`
Removes the widget w from the tab order list for e.
- `void ewl_embed_thaw (Ewl_Embed *e)`
Thaw redraws and events on the specified embed.
- `Ewl_Embed * ewl_embed_widget_find (Ewl_Widget *w)`
Find an ewl embed by a widget inside.
- `void ewl_embed_window_position_get (Ewl_Embed *e, int *x, int *y)`
Retrieve the position of the window.

6.20.3 Define Documentation

6.20.3.1 `#define EWL_EMBED(widget) ((Ewl_Embed *) widget)`

Typecast a pointer to an `Ewl_Embed` (p. 629) pointer.

6.20.3.2 `#define EWL_EMBED_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_EMBED_TYPE))`

Returns TRUE if the widget is an `Ewl_Embed` (p. 629), FALSE otherwise

6.20.3.3 `#define EWL_EMBED_TYPE "embed"`

The type name for the `Ewl_Embed` (p. 629) widget

6.20.3.4 `#define EWL_EMBED_WINDOW(window) ((Ewl_Embed_Window *)window)`

Typecast a pointer to an canvas window.

6.20.4 Typedef Documentation

6.20.4.1 `typedef struct Ewl_Embed Ewl_Embed`

The embed structure is mostly a container for holding widgets and a wrapper canvas smart object.

6.20.4.2 `typedef void* Ewl_Embed_Window`

Typedef and abstract out embedded canvas windows

6.20.5 Function Documentation

6.20.5.1 Ewl_Embed* ewl_embed_active_embed_get (void)

Returns the currently active embed widget.

Returns:

Returns the currently active embed

6.20.5.2 void ewl_embed_active_set (Ewl_Embed * *embed*, unsigned int *act*)

Sets if the given embed *embed* is active or not.

Parameters:

embed,: The embed to set the active value on

act,: The active value to set

Returns:

Returns no value.

6.20.5.3 void ewl_embed_cache_cleanup (Ewl_Embed * *emb*)

This will destroy the object cache associated with the embed object.

Parameters:

emb,: The embed to work with

Returns:

Returns no value

6.20.5.4 void* ewl_embed_canvas_set (Ewl_Embed * *emb*, void * *canvas*, Ewl_Embed_Window * *canvas_window*)

Change the canvas used by the embedded container.

Parameters:

emb,: the embedded container to change the target canvas

canvas,: the new canvas to draw the container and it's contents

canvas_window,: the window containing the canvas, for event dispatching

Returns:

Returns an canvas object on success, NULL on failure.

The returned smart object can be used to manipulate the area used by EWL through standard canvas functions.

6.20.5.5 Ewl_Embed* ewl_embed_canvas_window_find (Ewl_Embed_Window * *window*)

Find an ewl embed by its evas window.

Parameters:

window,: the evas window to search for on the list of embeds

Returns:

Returns the found embed on success, NULL on failure.

6.20.5.6 void ewl_embed_cb_configure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.20.5.7 void ewl_embed_cb_destroy (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.20.5.8 void ewl_embed_cb_focus_out (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.20.5.9 void ewl_embed_cb_realize (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.20.5.10 void ewl_embed_cb_unrealize (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.20.5.11 void ewl_embed_coord_to_screen (Ewl_Embed * *e*, int *xx*, int *yy*, int * *x*, int * *y*)

Maps coordinates from the Evas to screen coordinates.

Parameters:

e,: the embed to retrieve coord screen position

xx,: the x coord to map to a screen position

yy,: the y coord to map to a screen position

x,: storage for the mapped screen x position

y,: storage for the mapped screen y position

Returns:

Returns no value.

6.20.5.12 void ewl_embed_desktop_size_get (Ewl_Embed * *e*, int * *w*, int * *h*)

Retrieve the size of the root window.

Parameters:

e,: the embed to query for window-position

w,: a pointer to the integer that should receive the width

h,: a pointer to the integer that should receive the height

Returns:

Returns no value.

Stores the root window size into the parameters *w* and *h*.

6.20.5.13 void ewl_embed_dnd_aware_remove (Ewl_Embed * *embed*)

Cancels an embed as being DND aware.

Parameters:

embed,: the embed to remove dnd aware

Returns:

Returns no value.

6.20.5.14 void ewl_embed_dnd_aware_set (Ewl_Embed * *embed*)

Set an embed as being DND aware.

Parameters:

embed,: the embed to set dnd aware

Returns:

Returns no value.

6.20.5.15 void ewl_embed_dnd_data_received_feed (Ewl_Embed * *embed*, char * *type*, void * *data*, unsigned int *len*, unsigned int *format*)

Sends the event for selection data received into an embed.

Parameters:

embed,: the embed where the selection data event is to occur

type,: The type to feed.

data,: a pointer to the data received that generated the event

len,: length of the data that generated the event

format,: The bit format of the data

Returns:

Returns no value.

6.20.5.16 `void ewl_embed_dnd_data_request_feed (Ewl_Embed * embed, void * handle, char * type)`

Sends the request event for selection data received into an embed.

Parameters:

embed,: the embed where the selection data request event is to occur

handle,: The DND handle to work with

type,: The type to feed

Returns:

Returns no value.

6.20.5.17 `const char* ewl_embed_dnd_drop_feed (Ewl_Embed * embed, int x, int y, int internal)`

Sends the event for a DND drop into an embed.

Parameters:

embed,: the embed where the DND drop event is to occur

x,: the x coordinate of the mouse drop

y,: the y coordinate of the mouse drop

internal,: Is this an internal drop?

Returns:

Returns the DND drop data type.

6.20.5.18 `const char* ewl_embed_dnd_position_feed (Ewl_Embed * embed, int x, int y, int * px, int * py, int * pw, int * ph)`

Sends the event for a DND position into an embed.

Parameters:

embed,: the embed where the DND position event is to occur

x,: the x coordinate of the mouse move

y,: the y coordinate of the mouse move

px,: Where to store the position x value

py,: Where to store the position y value

pw,: Where to store the position width

ph,: Where to store the position height

Returns:

Returns the type of the DND data

6.20.5.19 `const char* ewl_embed_engine_name_get (Ewl_Embed * embed)`

Get the current engine on an embed.

Parameters:

embed,: Embed to get engine name

Returns:

Returns the name of the engine to use for the embed.

6.20.5.20 `int ewl_embed_engine_name_set (Ewl_Embed * embed, const char * engine)`

Changes the current engine on an embed to the specified engine.

Parameters:

embed,: Embed to change engines

engine,: Name of the new engine to use for the embed.

Returns:

Returns TRUE on success, FALSE on failure.

6.20.5.21 `int ewl_embed_focus_get (Ewl_Embed * embed)`

Retrieve the boolean value that indicates if the embed takes focus.

Parameters:

embed,: the embed to indicate if it takes focus

Returns:

Returns a boolean indicating if the embed takes focus.

6.20.5.22 `void ewl_embed_focus_set (Ewl_Embed * embed, int f)`

Sets the boolean flag in the embed to determine if it takes focus.

Parameters:

embed,: the embed to indicate if it takes focus

f,: boolean to determine if the embed takes focus

Returns:

Returns no value.

6.20.5.23 Ewl_Widget* ewl_embed_focused_widget_get (Ewl_Embed * *embed*)

Get the current focused widget in the embed.

Parameters:

embed,: The embed to get the currently focused widget

Returns:

Returns the currently focused widget

6.20.5.24 void ewl_embed_focused_widget_set (Ewl_Embed * *embed*, Ewl_Widget * *w*)

Set the current focused widget in the embed.

Parameters:

embed,: The embed to set the focused widget too

w,: The widget to set as having focus

Returns:

Returns no value.

6.20.5.25 void ewl_embed_font_path_add (char * *path*)

Add a font path to all embeds after realized.

Parameters:

path,: the font path to add to the embeds

Returns:

Returns no value.

Adds the search path to the evases created in the embeds. Using ewl_theme_font_path_add is preferred.

6.20.5.26 void ewl_embed_freeze (Ewl_Embed * *e*)

Freeze redraws and events on the specified embed.

Parameters:

e,: the embed to freeze redraws and canvas events

Returns:

Returns no value.

6.20.5.27 void ewl_embed_info_widgets_cleanup (Ewl_Embed * *e*, Ewl_Widget * *w*)

This will check to see if the given widget is one of the last selected, clicked, mouse_in or the drag-n-drop widget and if so, set them to NULL.

Parameters:

e,: The embed to cleanup

w,: The widget to check while cleaning up

Returns:

Returns no value.

6.20.5.28 int ewl_embed_init (Ewl_Embed * *w*)

initialize a embed to default values and callbacks

Parameters:

w,: the embed to be initialized to default values and callbacks

Returns:

Returns TRUE or FALSE depending on if initialization succeeds.

Sets the values and callbacks of a embed *w* to their defaults.

6.20.5.29 void ewl_embed_key_down_feed (Ewl_Embed * *embed*, const char * *keyname*, unsigned int *mods*)

Sends the event for a key press into an embed.

Parameters:

embed,: the embed where the key event is to occur

keyname,: the key press to trigger

mods,: the mask of key modifiers currently pressed

Returns:

Returns no value.

6.20.5.30 void ewl_embed_key_up_feed (Ewl_Embed * *embed*, const char * *keyname*, unsigned int *mods*)

Sends the event for a key release into an embed.

Parameters:

embed,: the embed where the key event is to occur

keyname,: the key release to trigger
mods,: the mask of key modifiers currently pressed

Returns:

Returns no value.

6.20.5.31 void ewl_embed_mouse_cursor_set (Ewl_Widget * w)

This will retrieve the attached cursor off of the given widget and display it. If there is no widget on the widget it will use the default cursor.

Parameters:

w,: The **Ewl_Widget** (p. 785) to get the pointer from

Returns:

Returns no value.

6.20.5.32 void ewl_embed_mouse_down_feed (Ewl_Embed * embed, int b, int clicks, int x, int y, unsigned int mods)

Sends the event for a mouse button press into an embed.

Parameters:

embed,: the embed where the mouse event is to occur
b,: the number of the button pressed
clicks,: the number of times the button was pressed
x,: the x coordinate of the mouse press
y,: the y coordinate of the mouse press
mods,: the mask of key modifiers currently pressed

Returns:

Returns no value.

6.20.5.33 void ewl_embed_mouse_move_feed (Ewl_Embed * embed, int x, int y, unsigned int mods)

Sends the event for a mouse button release into an embed.

Parameters:

embed,: the embed where the mouse event is to occur
x,: the x coordinate of the mouse move
y,: the y coordinate of the mouse move
mods,: the mask of key modifiers currently release

Returns:

Returns no value.

6.20.5.34 void ewl_embed_mouse_out_feed (Ewl_Embed * *embed*, int *x*, int *y*, unsigned int *mods*)

Sends a mouse out event to the last focused widget.

Parameters:

embed,: the embed where the mouse event is to occur
x,: the x coordinate of the mouse out
y,: the y coordinate of the mouse out
mods,: the mask of key modifiers currently release

Returns:

Returns no value.

6.20.5.35 void ewl_embed_mouse_up_feed (Ewl_Embed * *embed*, int *b*, int *x*, int *y*, unsigned int *mods*)

Sends the event for a mouse button release into an embed.

Parameters:

embed,: the embed where the mouse event is to occur
b,: the number of the button released
x,: the x coordinate of the mouse release
y,: the y coordinate of the mouse release
mods,: the mask of key modifiers currently release

Returns:

Returns no value.

6.20.5.36 void ewl_embed_mouse_wheel_feed (Ewl_Embed * *embed*, int *x*, int *y*, int *z*, int *dir*, unsigned int *mods*)

Sends a mouse out event to the last focused widget.

Parameters:

embed,: the embed where the mouse event is to occur
x,: the x coordinate of the mouse when the wheel event occurred
y,: the y coordinate of the mouse when wheel event occurred
z,: the direction the mouse wheel rotated
dir,: FIXME, not sure what this does, it's always 0
mods,: the mask of key modifiers currently release

Returns:

Returns no value.

6.20.5.37 Ewl_Widget* ewl_embed_new (void)

Allocate and initialize a new embed.

Returns:

Returns a new embed on success, or NULL on failure.

6.20.5.38 void ewl_embed_object_cache (Ewl_Embed * *e*, void * *obj*)

Caches the specified object for later reuse.

Parameters:

e,: embed to cache the specified object
obj,: the object to keep cached for reuse

Returns:

Returns no value.

6.20.5.39 void* ewl_embed_object_request (Ewl_Embed * *e*, char * *type*)

Retrieves an object of type *type* from the embed cache, or NULL if none found.

Parameters:

e,: embed to request a cached object
type,: the type of object requested

Returns:

Returns an Evas_Object of the specified type on success.

6.20.5.40 void ewl_embed_position_get (Ewl_Embed * *e*, int * *x*, int * *y*)

Retrieve the position of the embed.

Parameters:

e,: the embed to query for position
x,: a pointer to the integer that should receive the x coordinate
y,: a pointer to the integer that should receive the y coordinate

Returns:

Returns no value.

Stores the embed position into the parameters *x* and *y*.

6.20.5.41 void ewl_embed_shutdown (void)

6.20.5.42 void ewl_embed_tab_order_append (Ewl_Embed * *e*, Ewl_Widget * *w*)

Moves the widget *w* to the end of the tab order.

Parameters:

e,: The embed that holds the widgets

w,: The widget to be append to the tab order

Returns:

Returns no value.

6.20.5.43 void ewl_embed_tab_order_insert (Ewl_Embed * *e*, Ewl_Widget * *w*, unsigned int *idx*)

Moves the given widget *w* to the position *idx*.

Parameters:

e,: The embed that holds the widgets

w,: The widget to insert into the tab order

idx,: The index to insert into

Returns:

Returns no value.

6.20.5.44 void ewl_embed_tab_order_insert_after (Ewl_Embed * *e*, Ewl_Widget * *w*, Ewl_Widget * *after*)

Inserts the *w* widget into the tab order after the *after* widget.

Parameters:

e,: The embed to insert the widget into the tab order

w,: The widget to insert into the tab order

after,: The widget to insert after

Returns:

Returns no value.

6.20.5.45 void ewl_embed_tab_order_insert_before (Ewl_Embed * *e*, Ewl_Widget * *w*, Ewl_Widget * *before*)

Inserts the *w* widget into the tab order before the *before* widget.

Parameters:

e,: The embed to insert the widget into the tab order

w,: The widget to insert into the tab order

before,: The widget to insert after

Returns:

Returns no value.

6.20.5.46 void ewl_embed_tab_order_next (Ewl_Embed * e)

Changes focus to the next widget in the circular tab order list.

Parameters:

e,: the embed to change focus of it's contained widgets

Returns:

Returns no value.

6.20.5.47 void ewl_embed_tab_order_prepend (Ewl_Embed * e, Ewl_Widget * w)

Moves the widget *w* to the front of the tab order list.

Parameters:

e,: the embed that holds widgets

w,: the widget that will be moved to the front of the tab order list

Returns:

Returns no value.

6.20.5.48 void ewl_embed_tab_order_previous (Ewl_Embed * e)

Changes focus to the next widget in the circular tab order list.

Parameters:

e,: the embed to change focus of it's contained widgets

Returns:

Returns no value.

6.20.5.49 `void ewl_embed_tab_order_remove (Ewl_EMBED * e, Ewl_Widget * w)`

Removes the widget *w* from the tab order list for *e*.

Parameters:

e,: the embed containing a widget to remove from the tab order

w,: the widget to remove from the tab order list

Returns:

Returns no value.

6.20.5.50 `void ewl_embed_thaw (Ewl_EMBED * e)`

Thaw redraws and events on the specified embed.

Parameters:

e,: the embed to thaw redraws and canvas events

Returns:

Returns no value.

6.20.5.51 `Ewl_EMBED* ewl_embed_widget_find (Ewl_Widget * w)`

Find an ewl embed by a widget inside.

Parameters:

w,: the widget to search for its embed

Returns:

Returns the found embed on success, NULL on failure.

6.20.5.52 `void ewl_embed_window_position_get (Ewl_EMBED * e, int * x, int * y)`

Retrieve the position of the window.

Parameters:

e,: the embed to query for window-position

x,: a pointer to the integer that should receive the x coordinate

y,: a pointer to the integer that should receive the y coordinate

Returns:

Returns no value.

Stores the window position into the parameters *x* and *y*.

6.21 Ewl_Entry: A text entry widget

6.21.1 Detailed Description

This widget provides all text entry needs, from single to multiline

Remarks:

Inherits from **Ewl_Text** (p. 757).

6.21.2 Tutorial

The **Ewl_Entry** (p. 635) provides a widget for editing single line or multi-line text. Since the entry widget inherits from **Ewl_Text** (p. 757), all operations that can be performed on text can be used on the entry, including programmatic control of formatting, selections and text contents.

```
Ewl_Widget *entry;  
entry = ewl_entry_new();  
ewl_text_text_set(EWL_TEXT(entry), "Text to display");  
ewl_widget_show(entry);
```

Almost all manipulation of the entry widget is performed through the **Ewl_Text** (p. 757) API, with some minor exceptions. Since the entry widget takes user input, it requires an API to enable or disable the editing capability.

```
void ewl_entry_editable_set(Ewl_Entry *e, unsigned int editable);  
unsigned int ewl_entry_editable_get(Ewl_Entry *e);
```

It is also important to be able to indicate if the text entry can contain multiple lines of text, as the reaction to the Enter/Return keys changes. When multiline is enabled, Enter will insert a carriage return in the text, otherwise it raises an EWL_CALLBACK_VALUE_CHANGED callback on the entry widget.

```
void ewl_entry_multiline_set(Ewl_Entry *e, unsigned int multiline);  
unsigned int ewl_entry_multiline_get(Ewl_Entry *e);
```

By inheriting from **Ewl_Text** (p. 757), the capabilities of **Ewl_Entry** (p. 635) are expanded to the full API available for text manipulation. This reduces the number of calls for the user to learn without sacrificing functionality.

Data Structures

- struct **Ewl_Entry**

*Inherits from **Ewl_Text** (p. 757) and extends to allow editing of the text.*

- struct **Ewl_Entry_Cursor**

*Inherit from **Ewl_Widget** (p. 785) and extend for the Entry cursor.*

Defines

- `#define EWL_ENTRY(entry) ((Ewl_Entry *) entry)`
- `#define EWL_ENTRY_CURSOR(cursor) ((Ewl_Entry_Cursor *) cursor)`
- `#define EWL_ENTRY_CURSOR_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_ENTRY_CURSOR_TYPE))`
- `#define EWL_ENTRY_CURSOR_TYPE "cursor"`
- `#define EWL_ENTRY_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_ENTRY_TYPE))`
- `#define EWL_ENTRY_TYPE "entry"`

Typedefs

- `typedef Ewl_Entry Ewl_Entry`
- `typedef Ewl_Entry_Cursor Ewl_Entry_Cursor`

Functions

- `void ewl_entry_cb_configure (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_entry_cb_disable (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_entry_cb_dnd_data (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_entry_cb_dnd_position (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_entry_cb_enable (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_entry_cb_focus_in (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_entry_cb_focus_out (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_entry_cb_key_down (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_entry_cb_mouse_down (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_entry_cb_mouse_move (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_entry_cb_mouse_up (Ewl_Widget *w, void *ev, void *data)`
- `int ewl_entry_cursor_init (Ewl_Entry_Cursor *c, Ewl_Entry *parent)`
*Initializes an **Ewl_Entry_Cursor** (p. 637) to default values.*
- `void ewl_entry_cursor_move_down (Ewl_Entry *e)`
Moves the cursor down one line from its current position.
- `void ewl_entry_cursor_move_left (Ewl_Entry *e)`
Moves the cursor to the left of its current position.
- `void ewl_entry_cursor_move_right (Ewl_Entry *e)`
Moves the cursor to the right of its current position.
- `void ewl_entry_cursor_move_up (Ewl_Entry *e)`
Moves the cursor up one line from its current position.
- `Ewl_Widget * ewl_entry_cursor_new (Ewl_Entry *parent)`
*Creates and initializes a new **Ewl_Entry_Cursor** (p. 637) widget.*
- `unsigned int ewl_entry_cursor_position_get (Ewl_Entry_Cursor *c)`
*Retrieves the current position of the **Ewl_Entry_Cursor** (p. 637).*

- void **ewl_entry_cursor_position_set** (**Ewl_Entry_Cursor** *c, unsigned int pos)
Set the current cursor position.
- void **ewl_entry_delete_left** (**Ewl_Entry** *e)
Deletes the character to the left of the cursor.
- void **ewl_entry_delete_right** (**Ewl_Entry** *e)
Deletes the character to the right of the cursor.
- unsigned int **ewl_entry_editable_get** (**Ewl_Entry** *e)
Retrieve if the entry is editable or not.
- void **ewl_entry_editable_set** (**Ewl_Entry** *e, unsigned int editable)
Set if the entry is editable or not.
- int **ewl_entry_init** (**Ewl_Entry** *e)
*Initializes an **Ewl_Entry** (p. 635) widget to default values.*
- unsigned int **ewl_entry_multiline_get** (**Ewl_Entry** *e)
Retrieve if the entry is multiline or not.
- void **ewl_entry_multiline_set** (**Ewl_Entry** *e, unsigned int multiline)
Set if the entry is multiline or not.
- **Ewl_Widget** * **ewl_entry_new** (void)
*Create and return a new **Ewl_Entry** (p. 635) widget.*
- unsigned int **ewl_entry_selection_clear** (**Ewl_Entry** *e)
Clear the current selection in the entry.

6.21.3 Define Documentation

6.21.3.1 #define EWL_ENTRY(entry) ((Ewl_Entry *) entry)

Typecast a pointer to an **Ewl_Entry** (p. 635)

6.21.3.2 #define EWL_ENTRY_CURSOR(cursor) ((Ewl_Entry_Cursor *) cursor)

Typecast a pointer to an **Ewl_Entry_Cursor** (p. 637) pointer

6.21.3.3 #define EWL_ENTRY_CURSOR_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_ENTRY_CURSOR_TYPE))

Returns TRUE if the widget is an **Ewl_Entry_Cursor** (p. 637), FALSE otherwise

6.21.3.4 #define EWL_ENTRY_CURSOR_TYPE "cursor"

The type name for the **Ewl_Entry_Cursor** (p. 637) widget

6.21.3.5 `#define EWL_ENTRY_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_ENTRY_TYPE))`

Returns TRUE if the widget is an **Ewl_Entry** (p. 635), FALSE otherwise

6.21.3.6 `#define EWL_ENTRY_TYPE "entry"`

The type name for the **Ewl_Entry** (p. 635) widget

6.21.4 Typedef Documentation

6.21.4.1 `typedef struct Ewl_Entry Ewl_Entry`

The **Ewl_Entry** (p. 635)

6.21.4.2 `typedef struct Ewl_Entry_Cursor Ewl_Entry_Cursor`

The **Ewl_Entry_Cursor** (p. 637) widget

6.21.5 Function Documentation

6.21.5.1 `void ewl_entry_cb_configure (Ewl_Widget * w, void * ev, void * data)`

6.21.5.2 `void ewl_entry_cb_disable (Ewl_Widget * w, void * ev, void * data)`

6.21.5.3 `void ewl_entry_cb_dnd_data (Ewl_Widget * w, void * ev, void * data)`

6.21.5.4 `void ewl_entry_cb_dnd_position (Ewl_Widget * w, void * ev, void * data)`

6.21.5.5 `void ewl_entry_cb_enable (Ewl_Widget * w, void * ev, void * data)`

6.21.5.6 `void ewl_entry_cb_focus_in (Ewl_Widget * w, void * ev, void * data)`

6.21.5.7 `void ewl_entry_cb_focus_out (Ewl_Widget * w, void * ev, void * data)`

6.21.5.8 `void ewl_entry_cb_key_down (Ewl_Widget * w, void * ev, void * data)`

6.21.5.9 `void ewl_entry_cb_mouse_down (Ewl_Widget * w, void * ev, void * data)`

6.21.5.10 `void ewl_entry_cb_mouse_move (Ewl_Widget * w, void * ev, void * data)`

6.21.5.11 `void ewl_entry_cb_mouse_up (Ewl_Widget * w, void * ev, void * data)`

6.21.5.12 `int ewl_entry_cursor_init (Ewl_Entry_Cursor * c, Ewl_Entry * parent)`

Initializes an **Ewl_Entry_Cursor** (p. 637) to default values.

Parameters:

c,: The **Ewl_Entry_Cursor** (p. 637) to initialize

parent,: The parent **Ewl_Text** (p. 757) widget

Returns:

Returns TRUE on success or FALSE on failure

6.21.5.13 void ewl_entry_cursor_move_down (Ewl_Entry * e)

Moves the cursor down one line from its current position.

Parameters:

e,: The **Ewl_Entry** (p. 635) to work with

Returns:

Returns no value.

6.21.5.14 void ewl_entry_cursor_move_left (Ewl_Entry * e)

Moves the cursor to the left of it's current position.

Parameters:

e,: The **Ewl_Entry** (p. 635) to work with

Returns:

Returns no value

6.21.5.15 void ewl_entry_cursor_move_right (Ewl_Entry * e)

Moves the cursor to the right of its current position.

Parameters:

e,: The **Ewl_Entry** (p. 635) to work with

Returns:

Returns no value.

6.21.5.16 void ewl_entry_cursor_move_up (Ewl_Entry * e)

Moves the cursor up one line from its current position.

Parameters:

e,: The **Ewl_Entry** (p. 635) to work with

Returns:

Returns no value.

6.21.5.17 Ewl_Widget* ewl_entry_cursor_new (Ewl_Entry * *parent*)

Creates and initializes a new **Ewl_Entry_Cursor** (p. 637) widget.

Parameters:

parent,: The parent **Ewl_Entry** (p. 635) widget

Returns:

Returns a new **Ewl_Entry_Cursor** (p. 637) on success or NULL on failure

6.21.5.18 unsigned int ewl_entry_cursor_position_get (Ewl_Entry_Cursor * *c*)

Retrieves the current position of the **Ewl_Entry_Cursor** (p. 637).

Parameters:

c,: The cursor to get the position from

Returns:

Returns the current position of the cursor

6.21.5.19 void ewl_entry_cursor_position_set (Ewl_Entry_Cursor * *c*, unsigned int *pos*)

Set the current cursor position.

Parameters:

c,: The cursor to work with

pos,: The position to set the cursor too

Returns:

Returns no value.

6.21.5.20 void ewl_entry_delete_left (Ewl_Entry * *e*)

Deletes the character to the left of the cursor.

Parameters:

e,: The **Ewl_Entry** (p. 635) to work with

Returns:

Returns no value.

6.21.5.21 void ewl_entry_delete_right (Ewl_Entry * e)

Deletes the character to the right of the cursor.

Parameters:

e,: The **Ewl_Entry** (p. 635) to work with

Returns:

Returns no value.

6.21.5.22 unsigned int ewl_entry_editable_get (Ewl_Entry * e)

Retrieve if the entry is editable or not.

Parameters:

e,: The **Ewl_Entry** (p. 635) to get the editable status from

Returns:

Returns the editable status of the entry

6.21.5.23 void ewl_entry_editable_set (Ewl_Entry * e, unsigned int *editable*)

Set if the entry is editable or not.

Parameters:

e,: The **Ewl_Entry** (p. 635) to set the editable status of
editable,: The value to set for the editable flag

Returns:

Returns no value

6.21.5.24 int ewl_entry_init (Ewl_Entry * e)

Initializes an **Ewl_Entry** (p. 635) widget to default values.

Parameters:

e,: The **Ewl_Entry** (p. 635) to initialize

Returns:

Returns TRUE on success or FALSE on failure

6.21.5.25 unsigned int ewl_entry_multiline_get (Ewl_Entry * e)

Retrieve if the entry is multiline or not.

Parameters:

e,: The **Ewl_Entry** (p. 635) to get the multiline status from

Returns:

Returns the multiline status of the widget

6.21.5.26 void ewl_entry_multiline_set (Ewl_Entry * e, unsigned int *multiline*)

Set if the entry is multiline or not.

Parameters:

e,: The **Ewl_Entry** (p. 635) to set the multiline status

multiline,: The multiline status to set

Returns:

Returns no value

6.21.5.27 Ewl_Widget* ewl_entry_new (void)

Create and return a new **Ewl_Entry** (p. 635) widget.

Returns:

Returns a new **Ewl_Widget** (p. 785) on success or NULL on failure

6.21.5.28 unsigned int ewl_entry_selection_clear (Ewl_Entry * e)

Clear the current selection in the entry.

Parameters:

e,: The entry to clear the selection of

Returns:

Returns TRUE if a selection was cleared, FALSE otherwise.

6.22 Ewl_Enums: Various Flags and Enumerations used in EWL

6.22.1 Detailed Description

Provides bitmasks, flags, and other enumerations for use by widgets in EWL.

Defines

- `#define EWL_FLAG_FILL_NORMAL (EWL_FLAG_FILL_FILL)`
- `#define EWL_FLAGS_ALIGN_MASK`
- `#define EWL_FLAGS_FILL_MASK`
- `#define EWL_FLAGS_PROPERTY_MASK`
- `#define EWL_FLAGS_QUEUED_MASK`
- `#define EWL_FLAGS_STATE_MASK`
- `#define EWL_FLAGS_VISIBLE_MASK`
- `#define EWL_POSITION_MASK (0xf)`

Typedefs

- `typedef enum Ewl_Attach_Data_Type Ewl_Attach_Data_Type`
- `typedef enum Ewl_Attach_Type Ewl_Attach_Type`
- `typedef enum Ewl_Callback_Type Ewl_Callback_Type`
- `typedef enum Ewl_Color_Mode Ewl_Color_Mode`
- `typedef enum Ewl_Event_Notify Ewl_Event_Notify`
- `typedef enum Ewl_Filelist_Event_Type Ewl_Filelist_Event_Type`
- `typedef enum Ewl_Freebox_Layout_Type Ewl_Freebox_Layout_Type`
- `typedef enum Ewl_Grid_Resize_Type Ewl_Grid_Resize_Type`
- `typedef enum Ewl_Histogram_Channel Ewl_Histogram_Channel`
- `typedef enum Ewl_Icon_Part Ewl_Icon_Part`
- `typedef enum Ewl_Icon_Type Ewl_Icon_Type`
- `typedef enum Ewl_Image_Type Ewl_Image_Type`
- `typedef enum Ewl_Key_Modifiers Ewl_Key_Modifiers`
- `typedef enum Ewl_Media_Module_Type Ewl_Media_Module_Type`
- `typedef enum Ewl_Mouse_Cursor_Type Ewl_Mouse_Cursor_Type`
- `typedef enum Ewl_Notebook_Flags Ewl_Notebook_Flags`
- `typedef enum Ewl_Orientation Ewl_Orientation`
- `typedef enum Ewl_Popup_Type Ewl_Popup_Type`
- `typedef enum Ewl_Position Ewl_Position`
- `typedef enum Ewl_Rotate Ewl_Rotate`
- `typedef enum Ewl_Scrollpane_Flags Ewl_Scrollpane_Flags`
- `typedef enum Ewl_Selection_Mode Ewl_Selection_Mode`
- `typedef enum Ewl_Selection_Type Ewl_Selection_Type`
- `typedef enum Ewl_Sort_Direction Ewl_Sort_Direction`
- `typedef enum Ewl_Spectrum_Type Ewl_Spectrum_Type`
- `typedef enum Ewl_State_Type Ewl_State_Type`
- `typedef enum Ewl_Stock_Type Ewl_Stock_Type`
- `typedef enum Ewl_Text_Context_Mask Ewl_Text_Context_Mask`
- `typedef enum Ewl_Text_Style Ewl_Text_Style`

- typedef enum Ewl_Text_Trigger_Type Ewl_Text_Trigger_Type
- typedef enum Ewl_Text_Wrap Ewl_Text_Wrap
- typedef enum Ewl_Tree_Node_Flags Ewl_Tree_Node_Flags
- typedef enum Ewl_Tree_Selection_Type Ewl_Tree_Selection_Type
- typedef enum Ewl_Window_Flags Ewl_Window_Flags

Enumerations

- enum Ewl_Attach_Data_Type { EWL_ATTACH_DATA_TYPE_TEXT, EWL_ATTACH_DATA_TYPE_WIDGET, EWL_ATTACH_DATA_TYPE_OTHER }
- enum Ewl_Attach_Type {
EWL_ATTACH_TYPE_TOOLTIP, EWL_ATTACH_TYPE_COLOR,
EWL_ATTACH_TYPE_NAME, EWL_ATTACH_TYPE_MOUSE_CURSOR,
EWL_ATTACH_TYPE_MOUSE_ARGB_CURSOR, EWL_ATTACH_TYPE_WIDGET_ASSOCIATION, EWL_ATTACH_TYPE_DND_DATA
}
- enum Ewl_Callback_Type {
EWL_CALLBACK_REALIZE, EWL_CALLBACK_UNREALIZE, EWL_CALLBACK_SHOW, EWL_CALLBACK_HIDE,
EWL_CALLBACK_OBSCURE, EWL_CALLBACK_REVEAL, EWL_CALLBACK_DESTROY, EWL_CALLBACK_CONFIGURE,
EWL_CALLBACK_REPARENT, EWL_CALLBACK_KEY_DOWN, EWL_CALLBACK_KEY_UP, EWL_CALLBACK_MOUSE_DOWN,
EWL_CALLBACK_MOUSE_UP, EWL_CALLBACK_MOUSE_MOVE, EWL_CALLBACK_MOUSE_WHEEL, EWL_CALLBACK_MOUSE_IN,
EWL_CALLBACK_MOUSE_OUT, EWL_CALLBACK_CLICKED, EWL_CALLBACK_FOCUS_IN, EWL_CALLBACK_FOCUS_OUT,
EWL_CALLBACK_VALUE_CHANGED, EWL_CALLBACK_STATE_CHANGED, EWL_CALLBACK_WIDGET_ENABLE, EWL_CALLBACK_WIDGET_DISABLE,
EWL_CALLBACK_MAX }
}
- enum Ewl_Color_Mode {
EWL_COLOR_MODE_RGB_RED, EWL_COLOR_MODE_RGB_GREEN, EWL_COLOR_MODE_RGB_BLUE, EWL_COLOR_MODE_HSV_HUE,
EWL_COLOR_MODE_HSV_SATURATION, EWL_COLOR_MODE_HSV_VALUE }
}
- enum Ewl_Event_Notify { EWL_CALLBACK_NOTIFY_NONE = 0, EWL_CALLBACK_NOTIFY_NOTIFY = 1, EWL_CALLBACK_NOTIFY_INTERCEPT = 2, EWL_CALLBACK_TYPE_DIRECT = 4 }
- enum Ewl_Filelist_Event_Type { EWL_FILELIST_EVENT_DIR_CHANGE, EWL_FILELIST_EVENT_SELECTION_CHANGE, EWL_FILELIST_EVENT_FILE_SELECTED }
- enum Ewl_Flags {
EWL_FLAG_ALIGN_CENTER = 0, EWL_FLAG_ALIGN_LEFT = 0x1,
EWL_FLAG_ALIGN_RIGHT = 0x2, EWL_FLAG_ALIGN_TOP = 0x4,
}

```

EWL_FLAG_ALIGN_BOTTOM = 0x8, EWL_FLAG_FILL_NONE = 0,
EWL_FLAG_FILL_HSHRINK = 0x10, EWL_FLAG_FILL_VSHRINK =
0x20,
EWL_FLAG_FILL_SHRINK, EWL_FLAG_FILL_HFILL = 0x40, EWL_
FLAG_FILL_VFILL = 0x80, EWL_FLAG_FILL_FILL = EWL_FLAG_FILL_
HFILL | EWL_FLAG_FILL_VFILL,
EWL_FLAG_FILL_ALL = EWL_FLAG_FILL_FILL | EWL_FLAG_FILL_
SHRINK, EWL_FLAG_VISIBLE_HIDDEN = 0, EWL_FLAG_VISIBLE_
SHOWN = 0x100, EWL_FLAG_VISIBLE_REALIZED = 0x200,
EWL_FLAG_VISIBLE_OBSCURED = 0x400, EWL_FLAG_VISIBLE_
NOCLIP = 0x800, EWL_FLAG_PROPERTY_RECURSIVE = 0x1000, EWL_
FLAG_PROPERTY_TOPLEVEL = 0x2000,
EWL_FLAG_PROPERTY_INTERNAL = 0x4000, EWL_FLAG_
PROPERTY_BLOCK_TAB_FOCUS = 0x8000, EWL_FLAG_PROPERTY_
FOCUSABLE = 0x10000, EWL_FLAG_PROPERTY_IN_TAB_LIST =
0x20000,
EWL_FLAG_PROPERTY_DND_TARGET = 0x40000, EWL_FLAG_
PROPERTY_DND_SOURCE = 0x80000, EWL_FLAG_QUEUED_
CSCHEDULED = 0x100000, EWL_FLAG_QUEUED_RSCHEDULED =
0x200000,
EWL_FLAG_QUEUED_DSCHEDULED = 0x400000, EWL_FLAG_
QUEUED_CPROCESS = 0x800000, EWL_FLAG_QUEUED_RPROCESS
= 0x1000000, EWL_FLAG_QUEUED_DPROCESS = 0x2000000,
EWL_FLAG_STATE_NORMAL = 0, EWL_FLAG_STATE_MOUSE_IN
= 0x4000000, EWL_FLAG_STATE_PRESSED = 0x8000000, EWL_FLAG_
STATE_FOCUSED = 0x10000000,
EWL_FLAG_STATE_DISABLED = 0x20000000, EWL_FLAG_STATE_DND
= 0x40000000, EWL_FLAG_STATE_DND_WAIT = 0x80000000 }
• enum Ewl_Freebox_Layout_Type { EWL_FREEBOX_LAYOUT_MANUAL,
EWL_FREEBOX_LAYOUT_COMPARATOR, EWL_FREEBOX_
LAYOUT_AUTO }
• enum Ewl_Grid_Resize_Type { EWL_GRID_RESIZE_NORMAL, EWL_
GRID_RESIZE_FIXED, EWL_GRID_RESIZE_RELATIVE, EWL_
GRID_RESIZE_NONE }
• enum Ewl_Histogram_Channel {
EWL_HISTOGRAM_CHANNEL_Y, EWL_HISTOGRAM_CHANNEL_R,
EWL_HISTOGRAM_CHANNEL_G, EWL_HISTOGRAM_CHANNEL_B,
EWL_HISTOGRAM_CHANNEL_MAX }
• enum Ewl_Icon_Part { EWL_ICON_PART_NONE, EWL_ICON_PART_
IMAGE, EWL_ICON_PART_LABEL }
• enum Ewl_Icon_Type { EWL_ICON_TYPE_SHORT, EWL_ICON_TYPE_
LONG }
• enum Ewl_Image_Type { EWL_IMAGE_TYPE_NORMAL, EWL_
IMAGE_TYPE_EDJE }
• enum Ewl_Key_Modifiers {
EWL_KEY_MODIFIER_SHIFT = 0x1, EWL_KEY_MODIFIER_CTRL =
0x2, EWL_KEY_MODIFIER_ALT = 0x4, EWL_KEY_MODIFIER_MOD =
0x8,
EWL_KEY_MODIFIER_WIN = 0x10 }

```

```

• enum Ewl_Media_Module_Type { EWL_MEDIA_MODULE_XINE, EWL_
  MEDIA_MODULE_GSTREAMER }
• enum Ewl_Mouse_Cursor_Type {
  EWL_MOUSE_CURSOR_X = 0, EWL_MOUSE_CURSOR_ARROW = 2,
  EWL_MOUSE_CURSOR_BASED_ARROW_DOWN = 4, EWL_MOUSE_
  CURSOR_UP = 6,
  EWL_MOUSE_CURSOR_BOAT = 8, EWL_MOUSE_CURSOR_
  BOTTOM_LEFT_CORNER = 12, EWL_MOUSE_CURSOR_BOTTOM_
  RIGHT_CORNER = 14, EWL_MOUSE_CURSOR_BOTTOM_SIDE =
  16,
  EWL_MOUSE_CURSOR_BOTTOM_TEE = 18, EWL_MOUSE_
  CURSOR_BOX_SPIRAL = 20, EWL_MOUSE_CURSOR_CENTER_PTR
  = 22, EWL_MOUSE_CURSOR_CIRCLE = 24,
  EWL_MOUSE_CURSOR_CLOCK = 26, EWL_MOUSE_CURSOR_
  COFFEE_MUG = 28, EWL_MOUSE_CURSOR_CROSS = 30, EWL_
  MOUSE_CURSOR_CROSS_REVERSE = 32,
  EWL_MOUSE_CURSOR_CROSSHAIR = 34, EWL_MOUSE_CURSOR_
  DIAMOND_CROSS = 36, EWL_MOUSE_CURSOR_DOT = 38, EWL_
  MOUSE_CURSOR_DOT_BOX_MASK = 40,
  EWL_MOUSE_CURSOR_DOUBLE_ARROW = 42, EWL_MOUSE_
  CURSOR_DRAFT_LARGE = 44, EWL_MOUSE_CURSOR_DRAFT_
  SMALL = 46, EWL_MOUSE_CURSOR_DRAPED_BOX = 48,
  EWL_MOUSE_CURSOR_EXCHANGE = 50, EWL_MOUSE_CURSOR_
  FLEUR = 52, EWL_MOUSE_CURSOR_GOBLER = 54, EWL_MOUSE_
  CURSOR_GUMBY = 56,
  EWL_MOUSE_CURSOR_HAND1 = 58, EWL_MOUSE_CURSOR_
  HAND2 = 60, EWL_MOUSE_CURSOR_HEART = 62, EWL_MOUSE_
  CURSOR_ICON = 64,
  EWL_MOUSE_CURSOR_IRON_CROSS = 66, EWL_MOUSE_CURSOR_
  LEFT_PTR = 68, EWL_MOUSE_CURSOR_LEFT_SIDE = 70, EWL_
  MOUSE_CURSOR_LEFT_TEE = 72,
  EWL_MOUSE_CURSOR_LEFTBUTTON = 74, EWL_MOUSE_
  CURSOR_LL_ANGLE = 76, EWL_MOUSE_CURSOR_LR_ANGLE =
  78, EWL_MOUSE_CURSOR_MAN = 80,
  EWL_MOUSE_CURSOR_MIDDLEBUTTON = 82, EWL_MOUSE_
  CURSOR_MOUSE = 84, EWL_MOUSE_CURSOR_PENCIL = 86, EWL_
  MOUSE_CURSOR_PIRATE = 88,
  EWL_MOUSE_CURSOR_PLUS = 90, EWL_MOUSE_CURSOR_
  QUESTION_ARROW = 92, EWL_MOUSE_CURSOR_RIGHT_PTR =
  94, EWL_MOUSE_CURSOR_RIGHT_SIDE = 96,
  EWL_MOUSE_CURSOR_RIGHT_TEE = 98, EWL_MOUSE_CURSOR_
  RIGHTBUTTON = 100, EWL_MOUSE_CURSOR_RTL_LOGO = 102,
  EWL_MOUSE_CURSOR_SAILBOAT = 104,
  EWL_MOUSE_CURSOR_SB_DOWN_ARROW = 106, EWL_MOUSE_
  CURSOR_SB_H_DOUBLE_ARROW = 108, EWL_MOUSE_CURSOR_
  SB_LEFT_ARROW = 110, EWL_MOUSE_CURSOR_SB_RIGHT_
  ARROW = 112,
  EWL_MOUSE_CURSOR_SB_UP_ARROW = 114, EWL_MOUSE_
  CURSOR_SB_V_DOUBLE_ARROW = 116, EWL_MOUSE_CURSOR_
  SHUTTLE = 118, EWL_MOUSE_CURSOR_SIZING = 120,

```

```

EWL_MOUSE_CURSOR_SPIDER = 122, EWL_MOUSE_CURSOR_-
SPRAYCAN = 124, EWL_MOUSE_CURSOR_STAR = 126, EWL_MOUSE_-
CURSOR_TARGET = 128,

EWL_MOUSE_CURSOR_TCROSS = 130, EWL_MOUSE_CURSOR_-
TOP_LEFT_ARROW = 132, EWL_MOUSE_CURSOR_TOP_LEFT_-
CORNER = 134, EWL_MOUSE_CURSOR_TOP_RIGHT_CORNER =
136,

EWL_MOUSE_CURSOR_TOP_SIDE = 138, EWL_MOUSE_CURSOR_-
TOP_TEE = 140, EWL_MOUSE_CURSOR_TREK = 142, EWL_MOUSE_-
CURSOR_UL_ANGLE = 144,

EWL_MOUSE_CURSOR_UMBRELLA = 146, EWL_MOUSE_CURSOR_-
UR_ANGLE = 148, EWL_MOUSE_CURSOR_WATCH = 150, EWL_-
MOUSE_CURSOR_XTERM = 152,

EWL_MOUSE_CURSOR_MAX = 153 }

• enum Ewl_Notebook_Flags { EWL_NOTEBOOK_FLAG_TABS_HIDDEN =
0x10 }

• enum Ewl_Orientation { EWL_ORIENTATION_HORIZONTAL, EWL_-
ORIENTATION_VERTICAL }

• enum Ewl_Popup_Type { EWL_POPUP_TYPE_NONE, EWL_POPUP_-
TYPE_MOUSE, EWL_POPUP_TYPE_MENU_HORIZONTAL, EWL_-
POPUP_TYPE_MENU_VERTICAL }

• enum Ewl_Position { EWL_POSITION_LEFT = 0x1, EWL_POSITION_-
RIGHT = 0x2, EWL_POSITION_TOP = 0x4, EWL_POSITION_BOTTOM
= 0x8 }

• enum Ewl_Rotate {

EWL_ROTATE_CW_90, EWL_ROTATE_180, EWL_ROTATE_CW_270,
EWL_ROTATE_CC_90,

EWL_ROTATE_CC_270 }

• enum Ewl_Scrollpane_Flags { EWL_SCROLLPANE_FLAG_NONE, EWL_-
SCROLLPANE_FLAG_AUTO_VISIBLE, EWL_SCROLLPANE_FLAG_-
ALWAYS_HIDDEN }

• enum Ewl_Selection_Mode { EWL_SELECTION_MODE_NONE, EWL_-
SELECTION_MODE_SINGLE, EWL_SELECTION_MODE_MULTI }

• enum Ewl_Selection_Type { EWL_SELECTION_TYPE_INDEX, EWL_-
SELECTION_TYPE_RANGE }

• enum Ewl_Sort_Direction { EWL_SORT_DIRECTION_NONE = 0,
EWL_SORT_DIRECTION_ASCENDING, EWL_SORT_DIRECTION_-
DESCENDING, EWL_SORT_DIRECTION_MAX }

• enum Ewl_Spectrum_Type { EWL_SPECTRUM_TYPE_SQUARE, EWL_-
SPECTRUM_TYPE_VERTICAL }

• enum Ewl_State_Type { EWL_STATE_TRANSIENT, EWL_STATE_-
PERSISTENT }

• enum Ewl_Stock_Type {

EWL_STOCK_APPLY = 0, EWL_STOCK_ARROW_DOWN, EWL_-
STOCK_ARROW_LEFT, EWL_STOCK_ARROW_RIGHT,

EWL_STOCK_ARROW_UP, EWL_STOCK_CANCEL, EWL_STOCK_-
FASTFORWARD, EWL_STOCK_HOME,

EWL_STOCK_OK, EWL_STOCK_OPEN, EWL_STOCK_PAUSE, EWL_-
STOCK_PLAY,

```

```

EWL_STOCK_QUIT, EWL_STOCK_REWIND, EWL_STOCK_SAVE,
EWL_STOCK_STOP,
EWL_STOCK_NONE }
• enum Ewl_Text_Context_Mask {
    EWL_TEXT_CONTEXT_MASK_NONE = 0x00, EWL_TEXT_CONTEXT_MASK_FONT = 0x01, EWL_TEXT_CONTEXT_MASK_SIZE = 0x02, EWL_TEXT_CONTEXT_MASK_STYLES = 0x04,
    EWL_TEXT_CONTEXT_MASK_ALIGN = 0x08, EWL_TEXT_CONTEXT_MASK_WRAP = 0x10, EWL_TEXT_CONTEXT_MASK_COLOR = 0x20, EWL_TEXT_CONTEXT_MASK_BG_COLOR = 0x40,
    EWL_TEXT_CONTEXT_MASK_GLOW_COLOR = 0x80, EWL_TEXT_CONTEXT_MASK_OUTLINE_COLOR = 0x100, EWL_TEXT_CONTEXT_MASK_SHADOW_COLOR = 0x200, EWL_TEXT_CONTEXT_MASK_STRIKETHROUGH_COLOR = 0x400,
    EWL_TEXT_CONTEXT_MASK_UNDERLINE_COLOR = 0x800, EWL_TEXT_CONTEXT_MASK_DOUBLE_UNDERLINE_COLOR = 0x1000 }
• enum Ewl_Text_Style {
    EWL_TEXT_STYLE_NONE = 0x00, EWL_TEXT_STYLE_UNDERLINE = 0x01, EWL_TEXT_STYLE_DOUBLE_UNDERLINE = 0x02, EWL_TEXT_STYLE_STRIKETHROUGH = 0x04,
    EWL_TEXT_STYLE_SHADOW = 0x08, EWL_TEXT_STYLE_SOFT_SHADOW = 0x10, EWL_TEXT_STYLE_FAR_SHADOW = 0x20, EWL_TEXT_STYLE_OUTLINE = 0x40,
    EWL_TEXT_STYLE_GLOW = 0x80 }
• enum Ewl_Text_Trigger_Type { EWL_TEXT_TRIGGER_TYPE_NONE, EWL_TEXT_TRIGGER_TYPE_SELECTION, EWL_TEXT_TRIGGER_TYPE_TRIGGER }
• enum Ewl_Text_Wrap { EWL_TEXT_WRAP_NONE = 0, EWL_TEXT_WRAP_CHAR, EWL_TEXT_WRAP_WORD }
• enum Ewl_Tree_Node_Flags { EWL_TREE_NODE_NOEXPAND = 0, EWL_TREE_NODE_COLLAPSED = 1, EWL_TREE_NODE_EXPANDED = 2 }
• enum Ewl_Tree_Selection_Type { EWL_TREE_SELECTION_TYPE_ROW, EWL_TREE_SELECTION_TYPE_CELL }
• enum Ewl_Window_Flags {
    EWL_WINDOW_BORDERLESS = 0x1, EWL_WINDOW_USER_CONFIGURE = 0x2, EWL_WINDOW_GRAB_POINTER = 0x4, EWL_WINDOW_GRAB_KEYBOARD = 0x8,
    EWL_WINDOW_OVERRIDE = 0x10, EWL_WINDOW_FULLSCREEN = 0x20, EWL_WINDOW_DIALOG = 0x40, EWL_WINDOW_MODAL = 0x80,
    EWL_WINDOW_TRANSIENT = 0x100, EWL_WINDOW_TRANSIENT_FOREIGN = 0x200, EWL_WINDOW_SKIP_TASKBAR = 0x400, EWL_WINDOW_SKIP_PAGER = 0x800,
    EWL_WINDOW_DEMANDS_ATTENTION = 0x1000 }

```

6.22.2 Define Documentation

6.22.2.1 #define EWL_FLAG_FILL_NORMAL (EWL_FLAG_FILL_FILL)

Flag specifying normal fill values

6.22.2.2 #define EWL_FLAGS_ALIGN_MASK**Value:**

```
(EWL_FLAG_ALIGN_CENTER | EWL_FLAG_ALIGN_LEFT | \  
    EWL_FLAG_ALIGN_RIGHT | EWL_FLAG_ALIGN_TOP | \  
    EWL_FLAG_ALIGN_BOTTOM)
```

The alignment mask

6.22.2.3 #define EWL_FLAGS_FILL_MASK**Value:**

```
(EWL_FLAG_FILL_NONE | EWL_FLAG_FILL_SHRINK | \  
    EWL_FLAG_FILL_FILL)
```

The fill mask

6.22.2.4 #define EWL_FLAGS_PROPERTY_MASK**Value:**

```
(EWL_FLAG_PROPERTY_RECURSIVE | \  
    EWL_FLAG_PROPERTY_TOPLEVEL | EWL_FLAG_PROPERTY_INTERNAL | \  
    EWL_FLAG_PROPERTY_BLOCK_TAB_FOCUS | EWL_FLAG_PROPERTY_FOCUSABLE | \  
    EWL_FLAG_PROPERTY_DND_TARGET | EWL_FLAG_PROPERTY_DND_SOURCE | \  
    EWL_FLAG_PROPERTY_IN_TAB_LIST)
```

The property mask

6.22.2.5 #define EWL_FLAGS_QUEUED_MASK**Value:**

```
(EWL_FLAG_QUEUED_CSCHEDULED | \  
    EWL_FLAG_QUEUED_RSCHEDULED | EWL_FLAG_QUEUED_DSCHEDULED | \  
    EWL_FLAG_QUEUED_CPROCESS | EWL_FLAG_QUEUED_RPROCESS | \  
    EWL_FLAG_QUEUED_DPROCESS)
```

The queue mask

6.22.2.6 #define EWL_FLAGS_STATE_MASK**Value:**

```
(EWL_FLAG_STATE_NORMAL | \  
    EWL_FLAG_STATE_MOUSE_IN | EWL_FLAG_STATE_PRESSED | \  
    EWL_FLAG_STATE_FOCUSED | EWL_FLAG_STATE_DND | \  
    EWL_FLAG_STATE_DISABLED | EWL_FLAG_STATE_DND_WAIT)
```

The state mask

6.22.2.7 #define EWL_FLAGS_VISIBLE_MASK

Value:

```
(EWL_FLAG_VISIBLE_HIDDEN | \  
    EWL_FLAG_VISIBLE_SHOWN | EWL_FLAG_VISIBLE_REALIZED | \  
    EWL_FLAG_VISIBLE_OBSCURED | EWL_FLAG_VISIBLE_NOCLIP)
```

The visibility mask

6.22.2.8 #define EWL_POSITION_MASK (0xf)

The mask for the Ewl_Position bitmap

6.22.3 Typedef Documentation**6.22.3.1 typedef enum Ewl_Attach_Data_Type Ewl_Attach_Data_Type**

The Ewl_Attach_Data_Type

6.22.3.2 typedef enum Ewl_Attach_Type Ewl_Attach_Type

The Ewl_Attach_Type

6.22.3.3 typedef enum Ewl_Callback_Type Ewl_Callback_Type

The Ewl_Callback_Type

6.22.3.4 typedef enum Ewl_Color_Mode Ewl_Color_Mode

The Ewl_Color_Mode type

6.22.3.5 typedef enum Ewl_Event_Notify Ewl_Event_Notify

The Ewl_Event_Notify type

6.22.3.6 typedef enum Ewl_Filelist_Event_Type Ewl_Filelist_Event_Type

The Ewl_Filelist_Event_Type

6.22.3.7 typedef enum Ewl_Freebox_Layout_Type Ewl_Freebox_Layout_Type

The Ewl_Freebox_Layout_Type

6.22.3.8 typedef enum Ewl_Grid_Resize_Type Ewl_Grid_Resize_Type

The Ewl_Grid_Resize_Type

6.22.3.9 typedef enum Ewl_Histogram_Channel Ewl_Histogram_Channel

Ewl_Histogram_Channel

6.22.3.10 typedef enum Ewl_Icon_Part Ewl_Icon_Part

The Ewl_Icon_Part

6.22.3.11 typedef enum Ewl_Icon_Type Ewl_Icon_Type

The Ewl_Icon_Type

6.22.3.12 typedef enum Ewl_Image_Type Ewl_Image_Type

The Ewl_Image_Type

6.22.3.13 typedef enum Ewl_Key_Modifiers Ewl_Key_Modifiers

The Ewl_Key_Modifiers type

6.22.3.14 typedef enum Ewl_Media_Module_Type Ewl_Media_Module_Type

The Ewl_Media_Module_Type

6.22.3.15 typedef enum Ewl_Mouse_Cursor_Type Ewl_Mouse_Cursor_Type

The Ewl_Mouse_Cursor_Type

6.22.3.16 typedef enum Ewl_Notebook_Flags Ewl_Notebook_Flags

The Ewl_Notebook_Flags type

6.22.3.17 typedef enum Ewl_Orientation Ewl_Orientation

The Ewl_Orientation type

6.22.3.18 typedef enum Ewl_Popup_Type Ewl_Popup_Type

The Ewl_Popup_Types type

6.22.3.19 typedef enum Ewl_Position Ewl_Position

The Ewl_Position type

6.22.3.20 typedef enum Ewl_Rotate Ewl_Rotate

The Ewl_Rotate

6.22.3.21 typedef enum Ewl_Scrollpane_Flags Ewl_Scrollpane_Flags

The Ewl_Scrollpane_Flags type

6.22.3.22 typedef enum Ewl_Selection_Mode Ewl_Selection_Mode

Ewl_Selection_Mode

6.22.3.23 typedef enum Ewl_Selection_Type Ewl_Selection_Type

Ewl_Selection_Type

6.22.3.24 typedef enum Ewl_Sort_Direction Ewl_Sort_Direction

The Ewl_Sort_Direction

6.22.3.25 typedef enum Ewl_Spectrum_Type Ewl_Spectrum_Type

The Ewl_Spectrum_Type type

6.22.3.26 typedef enum Ewl_State_Type Ewl_State_Type

The Ewl_State_Type type

6.22.3.27 typedef enum Ewl_Stock_Type Ewl_Stock_Type

The Ewl_Stock_Type type

6.22.3.28 typedef enum Ewl_Text_Context_Mask Ewl_Text_Context_Mask

The Ewl_Text_Context_Mask type

6.22.3.29 typedef enum Ewl_Text_Style Ewl_Text_Style

The Ewl_Text_Style type

6.22.3.30 typedef enum Ewl_Text_Trigger_Type Ewl_Text_Trigger_Type

The Ewl_Text_Trigger_Type

6.22.3.31 typedef enum Ewl_Text_Wrap Ewl_Text_Wrap

The Ewl_Text_Wrap

6.22.3.32 typedef enum Ewl_Tree_Node_Flags Ewl_Tree_Node_Flags

The Ewl_Tree_Node_Flags type

6.22.3.33 typedef enum Ewl_Tree_Selection_Type Ewl_Tree_Selection_Type

Ewl_Tree_Selection_Type

6.22.3.34 typedef enum Ewl_Window_Flags Ewl_Window_Flags

The Ewl_Window_Flags type

6.22.4 Enumeration Type Documentation**6.22.4.1 enum Ewl_Attach_Data_Type**

The types of data that can be set into a widgets attachment

Enumerator:

```
EWL_ATTACH_DATA_TYPE_TEXT
EWL_ATTACH_DATA_TYPE_WIDGET
EWL_ATTACH_DATA_TYPE_OTHER
```

6.22.4.2 enum Ewl_Attach_Type

The types that can be attached to a widget

Enumerator:

```
EWL_ATTACH_TYPE_TOOLTIP
EWL_ATTACH_TYPE_COLOR
EWL_ATTACH_TYPE_NAME
EWL_ATTACH_TYPE_MOUSE_CURSOR
EWL_ATTACH_TYPE_MOUSE_ARGB_CURSOR
EWL_ATTACH_TYPE_WIDGET_ASSOCIATION
EWL_ATTACH_TYPE_DND_DATA
```

6.22.4.3 enum Ewl_Callback_Type

This defines the various types of callbacks that can be hooked up for each widget.

Enumerator:

```
EWL_CALLBACK_REALIZE Event when a widget is first drawn
EWL_CALLBACK_UNREALIZE When a widget is no longer drawn
EWL_CALLBACK_SHOW A widget has been marked visible
EWL_CALLBACK_HIDE A widget is marked hidden
EWL_CALLBACK_OBSCURE Widget is offscreen
EWL_CALLBACK_REVEAL Widget returned to screen
EWL_CALLBACK_DESTROY The widget is freed
```

EWL_CALLBACK_CONFIGURE The object is being resized
EWL_CALLBACK_REPARENT A widget has been placed in a container
EWL_CALLBACK_KEY_DOWN A key was pressed down
EWL_CALLBACK_KEY_UP A key was released
EWL_CALLBACK_MOUSE_DOWN Mouse was pressed down
EWL_CALLBACK_MOUSE_UP Mouse was released
EWL_CALLBACK_MOUSE_MOVE Mouse was moved
EWL_CALLBACK_MOUSE_WHEEL Mouse wheel scrolled
EWL_CALLBACK_MOUSE_IN Mouse was placed over the widget
EWL_CALLBACK_MOUSE_OUT Mouse was moved away from the widget
EWL_CALLBACK_CLICKED Mouse was pressed and released on a widget
EWL_CALLBACK_FOCUS_IN Widget was selected by mouse or key
EWL_CALLBACK_FOCUS_OUT Widget was deselected by mouse or key
EWL_CALLBACK_VALUE_CHANGED Value in widget changed
EWL_CALLBACK_STATE_CHANGED Alter the state of the appearance
EWL_CALLBACK_WIDGET_ENABLE Widget has been re-enabled
EWL_CALLBACK_WIDGET_DISABLE Widget no longer takes input
EWL_CALLBACK_MAX Flag to indicate last value

6.22.4.4 enum Ewl_Color_Mode

The different colour picker modes

Enumerator:

EWL_COLOR_MODE_RGB_RED
EWL_COLOR_MODE_RGB_GREEN
EWL_COLOR_MODE_RGB_BLUE
EWL_COLOR_MODE_HSV_HUE
EWL_COLOR_MODE_HSV_SATURATION
EWL_COLOR_MODE_HSV_VALUE

6.22.4.5 enum Ewl_Event_Notify

Flags for the callbacks to indicate interception or notification of the parent.

Enumerator:

EWL_CALLBACK_NOTIFY_NONE No notification or interception
EWL_CALLBACK_NOTIFY_NOTIFY Notify of call
EWL_CALLBACK_NOTIFY_INTERCEPT Intercept call
EWL_CALLBACK_TYPE_DIRECT Direct call

6.22.4.6 enum Ewl_Filelist_Event_Type

The possible events that can happen from the filelist

Enumerator:

```
EWL_FILELIST_EVENT_DIR_CHANGE
EWL_FILELIST_EVENT_SELECTION_CHANGE
EWL_FILELIST_EVENT_FILE_SELECTED
```

6.22.4.7 enum Ewl_Flags

A variety of flags that affect layout, visibility, scheduling and properties of objects.

Enumerator:

```
EWL_FLAG_ALIGN_CENTER Center align bit
EWL_FLAG_ALIGN_LEFT Left align bit
EWL_FLAG_ALIGN_RIGHT Right align bit
EWL_FLAG_ALIGN_TOP Top align bit
EWL_FLAG_ALIGN_BOTTOM Bottom align bit
EWL_FLAG_FILL_NONE Do not fill or shrink in any direction
EWL_FLAG_FILL_HSHRINK Horizontally shrink bit
EWL_FLAG_FILL_VSHRINK Horizontally shrink bit
EWL_FLAG_FILL_SHRINK Shrink bit
EWL_FLAG_FILL_HFILL Horizontal fill bit
EWL_FLAG_FILL_VFILL Vertical fill bit
EWL_FLAG_FILL_FILL Fill bit
EWL_FLAG_FILL_ALL Shrunk and fill bit
EWL_FLAG_VISIBLE_HIDDEN Widget hidden
EWL_FLAG_VISIBLE_SHOWN Widget shown
EWL_FLAG_VISIBLE_REALIZED Widget realized
EWL_FLAG_VISIBLE_OBSCURED Widget obscured
EWL_FLAG_VISIBLE_NOCLIP Widget has no clip
EWL_FLAG_PROPERTY_RECURSIVE Widget is recursive
EWL_FLAG_PROPERTY_TOPLEVEL Widget is a top level widget
EWL_FLAG_PROPERTY_INTERNAL Widget is internal
EWL_FLAG_PROPERTY_BLOCK_TAB_FOCUS Widget will block tab focus
changes
EWL_FLAG_PROPERTY_FOCUSABLE Widget is focusable
EWL_FLAG_PROPERTY_IN_TAB_LIST Widget is in the tab order
EWL_FLAG_PROPERTY_DND_TARGET Widget is DND aware
EWL_FLAG_PROPERTY_DND_SOURCE Widget is draggable
EWL_FLAG_QUEUED_CSCHEDULED Configure scheduled
EWL_FLAG_QUEUED_RSCHEDULED Reveal scheduled
```

EWL_FLAG_QUEUED_DSCHEDULED Delete scheduled
EWL_FLAG_QUEUED_CPROCESS Configure in progress
EWL_FLAG_QUEUED_RPROCESS Reveal in progress
EWL_FLAG_QUEUED_DPROCESS Delete in progress
EWL_FLAG_STATE_NORMAL Widget state normal
EWL_FLAG_STATE_MOUSE_IN Mouse is in the widget
EWL_FLAG_STATE_PRESSED Widget is pressed
EWL_FLAG_STATE_FOCUSED Widget has focus
EWL_FLAG_STATE_DISABLED Widget is disabled
EWL_FLAG_STATE_DND Widget is engaged in DND
EWL_FLAG_STATE_DND_WAIT Widget is in DND wait state

6.22.4.8 enum Ewl_Freebox_Layout_Type

The possibly layout settings for the freebox

Enumerator:

EWL_FREEBOX_LAYOUT_MANUAL
EWL_FREEBOX_LAYOUT_COMPARATOR
EWL_FREEBOX_LAYOUT_AUTO

6.22.4.9 enum Ewl_Grid_Resize_Type

The different ways of resizing a column or a row

Enumerator:

EWL_GRID_RESIZE_NORMAL use the standard resize methode
EWL_GRID_RESIZE_FIXED use the user set size
EWL_GRID_RESIZE_RELATIVE use the user set relative size
EWL_GRID_RESIZE_NONE use the preferred size of the row

6.22.4.10 enum Ewl_Histogram_Channel

The histogram channels

Enumerator:

EWL_HISTOGRAM_CHANNEL_Y
EWL_HISTOGRAM_CHANNEL_R
EWL_HISTOGRAM_CHANNEL_G
EWL_HISTOGRAM_CHANNEL_B
EWL_HISTOGRAM_CHANNEL_MAX

6.22.4.11 enum Ewl_Icon_Part

The different Parts of an Icon

Enumerator:

EWL_ICON_PART_NONE
EWL_ICON_PART_IMAGE
EWL_ICON_PART_LABEL

6.22.4.12 enum Ewl_Icon_Type

The possible icon types

Enumerator:

EWL_ICON_TYPE_SHORT
EWL_ICON_TYPE_LONG

6.22.4.13 enum Ewl_Image_Type

The type of image

Enumerator:

EWL_IMAGE_TYPE_NORMAL Standard image type
EWL_IMAGE_TYPE_EDJE Edje image type

6.22.4.14 enum Ewl_Key_Modifiers

The key modifiers

Enumerator:

EWL_KEY_MODIFIER_SHIFT Shift key
EWL_KEY_MODIFIER_CTRL Ctrl key
EWL_KEY_MODIFIER_ALT Alt key
EWL_KEY_MODIFIER_MOD Mod key
EWL_KEY_MODIFIER_WIN Win key

6.22.4.15 enum Ewl_Media_Module_Type

The possible modules to use with ewl_media

Enumerator:

EWL_MEDIA_MODULE_XINE
EWL_MEDIA_MODULE_GSTREAMER

6.22.4.16 enum Ewl_Mouse_Cursor_Type

The possible mouse cursor settings

Enumerator:

```
EWL_MOUSE_CURSOR_X  
EWL_MOUSE_CURSOR_ARROW  
EWL_MOUSE_CURSOR_BASED_ARROW_DOWN  
EWL_MOUSE_CURSOR_UP  
EWL_MOUSE_CURSOR_BOAT  
EWL_MOUSE_CURSOR_BOTTOM_LEFT_CORNER  
EWL_MOUSE_CURSOR_BOTTOM_RIGHT_CORNER  
EWL_MOUSE_CURSOR_BOTTOM_SIDE  
EWL_MOUSE_CURSOR_BOTTOM_TEE  
EWL_MOUSE_CURSOR_BOX_SPIRAL  
EWL_MOUSE_CURSOR_CENTER_PTR  
EWL_MOUSE_CURSOR_CIRCLE  
EWL_MOUSE_CURSOR_CLOCK  
EWL_MOUSE_CURSOR_COFFEE_MUG  
EWL_MOUSE_CURSOR_CROSS  
EWL_MOUSE_CURSOR_CROSS_REVERSE  
EWL_MOUSE_CURSOR_CROSSHAIR  
EWL_MOUSE_CURSOR_DIAMOND_CROSS  
EWL_MOUSE_CURSOR_DOT  
EWL_MOUSE_CURSOR_DOT_BOX_MASK  
EWL_MOUSE_CURSOR_DOUBLE_ARROW  
EWL_MOUSE_CURSOR_DRAFT_LARGE  
EWL_MOUSE_CURSOR_DRAFT_SMALL  
EWL_MOUSE_CURSOR_DRAPED_BOX  
EWL_MOUSE_CURSOR_EXCHANGE  
EWL_MOUSE_CURSOR_FLEUR  
EWL_MOUSE_CURSOR_GOBLER  
EWL_MOUSE_CURSOR_GUMBY  
EWL_MOUSE_CURSOR_HAND1  
EWL_MOUSE_CURSOR_HAND2  
EWL_MOUSE_CURSOR_HEART  
EWL_MOUSE_CURSOR_ICON  
EWL_MOUSE_CURSOR_IRON_CROSS  
EWL_MOUSE_CURSOR_LEFT_PTR  
EWL_MOUSE_CURSOR_LEFT_SIDE  
EWL_MOUSE_CURSOR_LEFT_TEE  
EWL_MOUSE_CURSOR_LEFTBUTTON
```

EWL_MOUSE_CURSOR_LL_ANGLE
EWL_MOUSE_CURSOR_LR_ANGLE
EWL_MOUSE_CURSOR_MAN
EWL_MOUSE_CURSOR_MIDDLEBUTTON
EWL_MOUSE_CURSOR_MOUSE
EWL_MOUSE_CURSOR_PENCIL
EWL_MOUSE_CURSOR_PIRATE
EWL_MOUSE_CURSOR_PLUS
EWL_MOUSE_CURSOR_QUESTION_ARROW
EWL_MOUSE_CURSOR_RIGHT_PTR
EWL_MOUSE_CURSOR_RIGHT_SIDE
EWL_MOUSE_CURSOR_RIGHT_TEE
EWL_MOUSE_CURSOR_RIGHTBUTTON
EWL_MOUSE_CURSOR_RTL_LOGO
EWL_MOUSE_CURSOR_SAILBOAT
EWL_MOUSE_CURSOR_SB_DOWN_ARROW
EWL_MOUSE_CURSOR_SB_H_DOUBLE_ARROW
EWL_MOUSE_CURSOR_SB_LEFT_ARROW
EWL_MOUSE_CURSOR_SB_RIGHT_ARROW
EWL_MOUSE_CURSOR_SB_UP_ARROW
EWL_MOUSE_CURSOR_SB_V_DOUBLE_ARROW
EWL_MOUSE_CURSOR_SHUTTLE
EWL_MOUSE_CURSOR_SIZING
EWL_MOUSE_CURSOR_SPIDER
EWL_MOUSE_CURSOR_SPRAYCAN
EWL_MOUSE_CURSOR_STAR
EWL_MOUSE_CURSOR_TARGET
EWL_MOUSE_CURSOR_TCROSS
EWL_MOUSE_CURSOR_TOP_LEFT_ARROW
EWL_MOUSE_CURSOR_TOP_LEFT_CORNER
EWL_MOUSE_CURSOR_TOP_RIGHT_CORNER
EWL_MOUSE_CURSOR_TOP_SIDE
EWL_MOUSE_CURSOR_TOP_TEE
EWL_MOUSE_CURSOR_TREK
EWL_MOUSE_CURSOR_UL_ANGLE
EWL_MOUSE_CURSOR_UMBRELLA
EWL_MOUSE_CURSOR_UR_ANGLE
EWL_MOUSE_CURSOR_WATCH
EWL_MOUSE_CURSOR_XTERM
EWL_MOUSE_CURSOR_MAX

6.22.4.17 enum Ewl_Notebook_Flags

States effecting the notebook

Enumerator:

EWL_NOTEBOOK_FLAG_TABS_HIDDEN Tabbar is hidden

6.22.4.18 enum Ewl_Orientation

The orientation enum is used in a few widgets to specify whether the widget should be laid out in a horizontal or vertical fashion.

Enumerator:

EWL_ORIENTATION_HORIZONTAL Horizontal layout

EWL_ORIENTATION_VERTICAL Vertical layout

6.22.4.19 enum Ewl_Popup_Type

The possible popup types

Enumerator:

EWL_POPUP_TYPE_NONE No popup type

EWL_POPUP_TYPE_MOUSE Popup off the mouse

EWL_POPUP_TYPE_MENU_HORIZONTAL Popup with horizontal data

EWL_POPUP_TYPE_MENU_VERTICAL Popup with vertical data

6.22.4.20 enum Ewl_Position

The possible placement positions

Enumerator:

EWL_POSITION_LEFT Left position

EWL_POSITION_RIGHT Right position

EWL_POSITION_TOP Top position

EWL_POSITION_BOTTOM Bottom position

6.22.4.21 enum Ewl_Rotate

The rotate values

Enumerator:

EWL_ROTATE_CW_90 90 degree clockwise rotation

EWL_ROTATE_180 180 degree rotation

EWL_ROTATE_CW_270 270 degree clockwise rotation

EWL_ROTATE_CC_90 90 degree counter-clockwise rotation

EWL_ROTATE_CC_270 270 degree counter-clockwise rotation

6.22.4.22 enum Ewl_Scrollpane_Flags

The possible scrollpane settings

Enumerator:

EWL_SCROLLPANE_FLAG_NONE No flags set
EWL_SCROLLPANE_FLAG_AUTO_VISIBLE Hide if possible
EWL_SCROLLPANE_FLAG_ALWAYS_HIDDEN Always hide

6.22.4.23 enum Ewl_Selection_Mode

The slecetion mode enum

Enumerator:

EWL_SELECTION_MODE_NONE No selection allowed
EWL_SELECTION_MODE_SINGLE Single select only
EWL_SELECTION_MODE_MULTI Multi select mode

6.22.4.24 enum Ewl_Selection_Type

The selection type enum

Enumerator:

EWL_SELECTION_TYPE_INDEX Single index
EWL_SELECTION_TYPE_RANGE Range of indices

6.22.4.25 enum Ewl_Sort_Direction

The current sort direction

Enumerator:

EWL_SORT_DIRECTION_NONE
EWL_SORT_DIRECTION_ASCENDING
EWL_SORT_DIRECTION_DESCENDING
EWL_SORT_DIRECTION_MAX

6.22.4.26 enum Ewl_Spectrum_Type

The spectrum types

Enumerator:

EWL_SPECTRUM_TYPE_SQUARE
EWL_SPECTRUM_TYPE_VERTICAL

6.22.4.27 enum Ewl_State_Type

The state type

Enumerator:

EWL_STATE_TRANSIENT
EWL_STATE_PERSISTENT

6.22.4.28 enum Ewl_Stock_Type

The different stock types

Enumerator:

EWL_STOCK_APPLY
EWL_STOCK_ARROW_DOWN
EWL_STOCK_ARROW_LEFT
EWL_STOCK_ARROW_RIGHT
EWL_STOCK_ARROW_UP
EWL_STOCK_CANCEL
EWL_STOCK_FASTFORWARD
EWL_STOCK_HOME
EWL_STOCK_OK
EWL_STOCK_OPEN
EWL_STOCK_PAUSE
EWL_STOCK_PLAY
EWL_STOCK_QUIT
EWL_STOCK_REWIND
EWL_STOCK_SAVE
EWL_STOCK_STOP
EWL_STOCK_NONE

6.22.4.29 enum Ewl_Text_Context_Mask

The possible context masks for the ewl_text

Enumerator:

EWL_TEXT_CONTEXT_MASK_NONE
EWL_TEXT_CONTEXT_MASK_FONT
EWL_TEXT_CONTEXT_MASK_SIZE
EWL_TEXT_CONTEXT_MASK_STYLES
EWL_TEXT_CONTEXT_MASK_ALIGN
EWL_TEXT_CONTEXT_MASK_WRAP
EWL_TEXT_CONTEXT_MASK_COLOR

EWL_TEXT_CONTEXT_MASK_BG_COLOR
EWL_TEXT_CONTEXT_MASK_GLOW_COLOR
EWL_TEXT_CONTEXT_MASK_OUTLINE_COLOR
EWL_TEXT_CONTEXT_MASK_SHADOW_COLOR
EWL_TEXT_CONTEXT_MASK_STRIKETHROUGH_COLOR
EWL_TEXT_CONTEXT_MASK_UNDERLINE_COLOR
EWL_TEXT_CONTEXT_MASK_DOUBLE_UNDERLINE_COLOR

6.22.4.30 enum Ewl_Text_Style

The possible styles that can be set into ewl_text widgets

Enumerator:

EWL_TEXT_STYLE_NONE
EWL_TEXT_STYLE_UNDERLINE
EWL_TEXT_STYLE_DOUBLE_UNDERLINE
EWL_TEXT_STYLE_STRIKETHROUGH
EWL_TEXT_STYLE_SHADOW
EWL_TEXT_STYLE_SOFT_SHADOW
EWL_TEXT_STYLE_FAR_SHADOW
EWL_TEXT_STYLE_OUTLINE
EWL_TEXT_STYLE_GLOW

6.22.4.31 enum Ewl_Text_Trigger_Type

The types of trigger that can be set

Enumerator:

EWL_TEXT_TRIGGER_TYPE_NONE
EWL_TEXT_TRIGGER_TYPE_SELECTION
EWL_TEXT_TRIGGER_TYPE_TRIGGER

6.22.4.32 enum Ewl_Text_Wrap

The text wrap setting

Enumerator:

EWL_TEXT_WRAP_NONE
EWL_TEXT_WRAP_CHAR
EWL_TEXT_WRAP_WORD

6.22.4.33 enum Ewl_Tree_Node_Flags

The available tree node states

Enumerator:

EWL_TREE_NODE_NOEXPAND Node does not expand
EWL_TREE_NODE_COLLAPSED Node is collapsed
EWL_TREE_NODE_EXPANDED Node is expanded

6.22.4.34 enum Ewl_Tree_Selection_Type

The tree selection type

Enumerator:

EWL_TREE_SELECTION_TYPE_ROW Row selection
EWL_TREE_SELECTION_TYPE_CELL Cell selection

6.22.4.35 enum Ewl_Window_Flags

The possible window states

Enumerator:

EWL_WINDOW_BORDERLESS Window is borderless
EWL_WINDOW_USER_CONFIGURE User configure
EWL_WINDOW_GRAB_POINTER Window grabs pointer
EWL_WINDOW_GRAB_KEYBOARD Window grabs keyboard
EWL_WINDOW_OVERRIDE Window override setting
EWL_WINDOW_FULLSCREEN Window is fullscreen
EWL_WINDOW_DIALOG Window has no max and min buttons
EWL_WINDOW_MODAL Window is modal
EWL_WINDOW_TRANSIENT Window is transient for
EWL_WINDOW_TRANSIENT_FOREIGN Window is transient for
EWL_WINDOW_SKIP_TASKBAR Window skips taskbar
EWL_WINDOW_SKIP_PAGER Window skips pager
EWL_WINDOW_DEMANDS_ATTENTION Window requires attention

6.23 Ewl_Events: Lower Level Event Handlers

6.23.1 Detailed Description

Defines the routines that dispatch the lower level events to EWL.

Data Structures

- struct **Ewl_Dnd_Types**
Provides type information about an external DND drag.
- struct **Ewl_Event_Action_Response**
Stores the response from a user action.
- struct **Ewl_Event_Dnd_Data_Received**
Provides information about dnd drop data.
- struct **Ewl_Event_Dnd_Data_Request**
Provides information about dnd data requests.
- struct **Ewl_Event_Dnd_Drop**
Provides information about dnd drops movement.
- struct **Ewl_Event_Dnd_Position**
Provides information about dnd drops movement.
- struct **Ewl_Event_Key**
Provides the basic information about key events.
- struct **Ewl_Event_Key_Down**
Provides clients with necessary information about the key press event.
- struct **Ewl_Event_Key_Up**
Provides clients with necessary information about the key release event.
- struct **Ewl_Event_Mouse**
Provides base mouse information.
- struct **Ewl_Event_Mouse_Down**
Provides information about the mouse down event.
- struct **Ewl_Event_Mouse_In**
Provides information about the mouse entering.
- struct **Ewl_Event_Mouse_Move**
Provides information about mouse movement.
- struct **Ewl_Event_Mouse_Out**
Provides information about the mouse leaving.

- struct **Ewl_Event_Mouse_Up**
Provides information about the mouse up event.
- struct **Ewl_Event_Mouse_Wheel**
Provides information about the mouse wheel scrolling.
- struct **Ewl_Event_Window_Expose**
Describes the canvas region that received an expose notification.

Typedefs

- typedef **Ewl_Dnd_Types** **Ewl_Dnd_Types**
- typedef **Ewl_Event_Action_Response** **Ewl_Event_Action_Response**
- typedef **Ewl_Event_Dnd_Data_Received** **Ewl_Event_Dnd_Data_Received**
- typedef **Ewl_Event_Dnd_Data_Request** **Ewl_Event_Dnd_Data_Request**
- typedef **Ewl_Event_Dnd_Drop** **Ewl_Event_Dnd_Drop**
- typedef **Ewl_Event_Dnd_Position** **Ewl_Event_Dnd_Position**
- typedef **Ewl_Event_Key** **Ewl_Event_Key**
- typedef **Ewl_Event_Key_Down** **Ewl_Event_Key_Down**
- typedef **Ewl_Event_Key_Up** **Ewl_Event_Key_Up**
- typedef **Ewl_Event_Mouse** **Ewl_Event_Mouse**
- typedef **Ewl_Event_Mouse_Down** **Ewl_Event_Mouse_Down**
- typedef **Ewl_Event_Mouse_In** **Ewl_Event_Mouse_In**
- typedef **Ewl_Event_Mouse_Move** **Ewl_Event_Mouse_Move**
- typedef **Ewl_Event_Mouse_Out** **Ewl_Event_Mouse_Out**
- typedef **Ewl_Event_Mouse_Up** **Ewl_Event_Mouse_Up**
- typedef **Ewl_Event_Mouse_Wheel** **Ewl_Event_Mouse_Wheel**
- typedef **Ewl_Event_Window_Expose** **Ewl_Event_Window_Expose**

Functions

- unsigned int **ewl_ev_modifiers_get** (void)
Retrieve the current mask of modifiers that are set.
- void **ewl_ev_modifiers_set** (unsigned int modifiers)
Sets the key modifiers to the given value.

6.23.2 Typedef Documentation

6.23.2.1 typedef struct Ewl_Dnd_Types Ewl_Dnd_Types

The **Ewl_Dnd_Types** (p. 628) type

6.23.2.2 typedef struct Ewl_Event_Action_Response Ewl_Event_Action_Response

The **Ewl_Event_Action_Response** (p. 639) type

6.23.2.3 `typedef struct Ewl_Event_Dnd_Data_Received
Ewl_Event_Dnd_Data_Received`

The `Ewl_Event_Dnd_Data_Received` (p. 640) type

6.23.2.4 `typedef struct Ewl_Event_Dnd_Data_Request
Ewl_Event_Dnd_Data_Request`

The `Ewl_Event_Dnd_Data_Requested` type

6.23.2.5 `typedef struct Ewl_Event_Dnd_Drop Ewl_Event_Dnd_Drop`

The `Ewl_Event_Dnd_Drop` (p. 642) type

6.23.2.6 `typedef struct Ewl_Event_Dnd_Position Ewl_Event_Dnd_Position`

The `Ewl_Event_Dnd_Position` (p. 643) type

6.23.2.7 `typedef struct Ewl_Event_Key Ewl_Event_Key`

The `Ewl_Event_Key` (p. 644) structure

6.23.2.8 `typedef struct Ewl_Event_Key_Down Ewl_Event_Key_Down`

The `Ewl_Event_Window_Key_Down` structure

6.23.2.9 `typedef struct Ewl_Event_Key_Up Ewl_Event_Key_Up`

The `Ewl_Event_Key_Up` (p. 646) structure

6.23.2.10 `typedef struct Ewl_Event_Mouse Ewl_Event_Mouse`

The `Ewl_Event_Mouse` (p. 647) structure

6.23.2.11 `typedef struct Ewl_Event_Mouse_Down Ewl_Event_Mouse_Down`

The `Ewl_Event_Mouse_Down` (p. 648) structure

6.23.2.12 `typedef struct Ewl_Event_Mouse_In Ewl_Event_Mouse_In`

The `Ewl_Event_Mouse_In` (p. 649) structure

6.23.2.13 `typedef struct Ewl_Event_Mouse_Move Ewl_Event_Mouse_Move`

The `Ewl_Event_Mouse_Move` (p. 650) structure

6.23.2.14 `typedef struct Ewl_Event_Mouse_Out Ewl_Event_Mouse_Out`

The `Ewl_Event_Mouse_Out` (p. 651) structure

6.23.2.15 `typedef struct Ewl_Event_Mouse_Up Ewl_Event_Mouse_Up`

The `Ewl_Event_Mouse_Up` (p. 652) structure

6.23.2.16 `typedef struct Ewl_Event_Mouse_Wheel Ewl_Event_Mouse_Wheel`

The `Ewl_Event_Mouse_Wheel` (p. 653) structure

6.23.2.17 `typedef struct Ewl_Event_Window_Expose Ewl_Event_Window_Expose`

The `Ewl_Event_Window_Expose` (p. 654) structure

6.23.3 Function Documentation**6.23.3.1** `unsigned int ewl_ev_modifiers_get (void)`

Retrieve the current mask of modifiers that are set.

Returns:

Returns the current mask of modifier keys.

6.23.3.2 `void ewl_ev_modifiers_set (unsigned int modifiers)`

Sets the key modifiers to the given value.

Parameters:

modifiers,: the key modifiers to set

Returns:

Returns no value.

6.24 Ewl_Expansion: The Expansion node for Tree2

6.24.1 Detailed Description

Defines a widget for tree2's expansion node

Remarks:

Inherits from **Ewl_Check** (p. 597).

Data Structures

- struct **Ewl_Expansion**

*Inherits from **Ewl_Check** (p. 597) and extends it to provide a expanded state.*

Defines

- #define **EWL_EXPANSION(c)** ((**Ewl_Expansion** *)c)
- #define **EWL_EXPANSION_IS(w)** (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_EXPANSION_TYPE**))
- #define **EWL_EXPANSION_TYPE** "expansion"

Typedefs

- typedef **Ewl_Expansion** **Ewl_Expansion**

Functions

- void **ewl_expansion_cb_reveal** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_expansion_cb_update_expandable** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_expansion_expandable_set** (**Ewl_Expansion** *c, int expandable)
Change the expandable status of the expansion.
- void **ewl_expansion_expanded_set** (**Ewl_Expansion** *c, int expanded)
Change the expanded status of the expansion.
- int **ewl_expansion_init** (**Ewl_Expansion** *c)
Initialize the members and callbacks of a expansion.
- int **ewl_expansion_is_expandable** (**Ewl_Expansion** *c)
Determine the expandable state of the expansion.
- int **ewl_expansion_is_expanded** (**Ewl_Expansion** *c)
Determine the expanded state of the expansion.
- **Ewl_Widget** * **ewl_expansion_new** (void)
Allocate and initialize a new expansion.

6.24.2 Define Documentation

6.24.2.1 `#define EWL_EXPANSION(c) ((Ewl_Expansion *)c)`

Typecasts a pointer to an **Ewl_Expansion** (p. 655) pointer.

6.24.2.2 `#define EWL_EXPANSION_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_EXPANSION_TYPE))`

Returns TRUE if the widget is an **Ewl_Expansion** (p. 655), FALSE otherwise

6.24.2.3 `#define EWL_EXPANSION_TYPE "expansion"`

The type name for the **Ewl_Expansion** (p. 655) widget

6.24.3 Typedef Documentation

6.24.3.1 `typedef struct Ewl_Expansion Ewl_Expansion`

A tree expansion handle.

6.24.4 Function Documentation

6.24.4.1 `void ewl_expansion_cb_reveal (Ewl_Widget * w, void * ev_data, void * user_data)`

6.24.4.2 `void ewl_expansion_cb_update_expandable (Ewl_Widget * w, void * ev_data, void * user_data)`

6.24.4.3 `void ewl_expansion_expandable_set (Ewl_Expansion * cb, int c)`

Change the expandable status of the expansion.

Parameters:

- cb*,: the expansion to change the status
- c*,: the new status of the expansion

Returns:

Returns no value.

Changes the expandable status of the expansion and updates it's appearance to reflect the change.

6.24.4.4 `void ewl_expansion_expanded_set (Ewl_Expansion * cb, int c)`

Change the expanded status of the expansion.

Parameters:

- cb*,: the expansion to change the status

c,: the new status of the expansion

Returns:

Returns no value.

Changes the expanded status of the expansion and updates it's appearance to reflect the change.

6.24.4.5 int ewl_expansion_init (Ewl_Expansion * *cb*)

Initialize the members and callbacks of a expansion.

Parameters:

cb,: the expansion to initialize

Returns:

Returns TRUE on success, FALSE on failure.

The internal structures and callbacks of the expansion are initialized to default values.

6.24.4.6 int ewl_expansion_is_expandable (Ewl_Expansion * *cb*)

Determine the expandable state of the expansion.

Parameters:

cb,: the expansion to examine for it's expandable state

Returns:

Returns TRUE if the expansion is expandable, FALSE if not.

6.24.4.7 int ewl_expansion_is_expanded (Ewl_Expansion * *cb*)

Determine the expanded state of the expansion.

Parameters:

cb,: the expansion to examine for it's expanded state

Returns:

Returns TRUE if the expansion is expanded, FALSE if not.

6.24.4.8 Ewl_Widget* ewl_expansion_new (void)

Allocate and initialize a new expansion.

Returns:

Returns the newly allocated expansion on success, NULL on failure.

6.25 Ewl_Filedialog: A Dialog For Picking Files

6.25.1 Detailed Description

A simple dialog to show a file selector widget

Remarks:

Inherits from **Ewl_Dialog** (p. 626).

Data Structures

- struct **Ewl_Filedialog**

*Inherits from **Ewl_Dialog** (p. 626) and extends to create a filedialog.*

Defines

- #define **EWL_FILEDIALOG**(fd) ((**Ewl_Filedialog** *) fd)
- #define **EWL_FILEDIALOG_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FILEDIALOG_TYPE))
- #define **EWL_FILEDIALOG_TYPE** "filedialog"

Typedefs

- typedef **Ewl_Filedialog** **Ewl_Filedialog**

Functions

- void **ewl_filedialog_cb_delete_window** (**Ewl_Widget** *w, void *ev_data, void *data)
- const char * **ewl_filedialog_directory_get** (**Ewl_Filedialog** *fd)
Retrieve the current filedialog path.
- void **ewl_filedialog_directory_set** (**Ewl_Filedialog** *fd, const char *dir)
Changes the current path of a filedialog.
- void **ewl_filedialog_filter_add** (**Ewl_Filedialog** *fd, const char *name, const char *filter)
Add the filter named name to the combo box in the filedialog.
- int **ewl_filedialog_init** (**Ewl_Filedialog** *fd)
Initialize a new filedialog.
- **Ewl_View** * **ewl_filedialog_list_view_get** (**Ewl_Filedialog** *fd)
Retrieve the file list view used in this file dialog.
- void **ewl_filedialog_list_view_set** (**Ewl_Filedialog** *fd, **Ewl_View** *view)
Set the view to be used for displaying the files in the dialog.

- unsigned int **ewl_filedialog_multiselect_get** (**Ewl_Filedialog** *fd)
gets the multiselect setting of the filedialog
- **Ewl_Widget** * **ewl_filedialog_multiselect_new** (void)
Create a new open filedialog.
- void **ewl_filedialog_multiselect_set** (**Ewl_Filedialog** *fd, unsigned int ms)
Sets the dialog to multiselect or single select.
- **Ewl_Widget** * **ewl_filedialog_new** (void)
Create a new filedialog.
- char * **ewl_filedialog_selected_file_get** (**Ewl_Filedialog** *fd)
Retrieve the selected filename.
- void **ewl_filedialog_selected_file_set** (**Ewl_Filedialog** *fd, const char *file)
Set the currently selected file into the file dialog.
- **Ecore_List** * **ewl_filedialog_selected_files_get** (**Ewl_Filedialog** *fd)
returns all the elements selected by the user
- void **ewl_filedialog_selected_files_set** (**Ewl_Filedialog** *fd, **Ecore_List** *files)
Sets the given files as selected in the filedialog.
- unsigned int **ewl_filedialog_show_dot_files_get** (**Ewl_Filedialog** *fd)
Returns the current dot file setting of the dialog.
- void **ewl_filedialog_show_dot_files_set** (**Ewl_Filedialog** *fd, unsigned int dot)
Set if the file dialog should show dot files by default.
- unsigned int **ewl_filedialog_show_favorites_get** (**Ewl_Filedialog** *fd)
Get the current show favorites setting for the filedialog.
- void **ewl_filedialog_show_favorites_set** (**Ewl_Filedialog** *fd, unsigned int show)
Specify if the favorites column should be shown or not.

6.25.2 Define Documentation

6.25.2.1 #define EWL_FILEDIALOG(fd) ((Ewl_Filedialog *) fd)

Typecasts a pointer to an **Ewl_Filedialog** (p. 656) pointer.

6.25.2.2 #define EWL_FILEDIALOG_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FILEDIALOG_TYPE))

Returns TRUE if the widget is an **Ewl_Filedialog** (p. 656), FALSE otherwise

6.25.2.3 `#define EWL_FILEDIALOG_TYPE "filedialog"`

The type name for the `Ewl_Filedialog` (p. 656) widget

6.25.3 Typedef Documentation

6.25.3.1 `typedef struct Ewl_Filedialog Ewl_Filedialog`

The `Ewl_Filedialog` (p. 656) provides a filedialog

6.25.4 Function Documentation

6.25.4.1 `void ewl_filedialog_cb_delete_window (Ewl_Widget * w, void * ev_data, void * data)`

6.25.4.2 `const char* ewl_filedialog_directory_get (Ewl_Filedialog * fd)`

Retrieve the current filedialog path.

Parameters:

fd,: the filedialog

Returns:

Returns the current path of filedialog

6.25.4.3 `void ewl_filedialog_directory_set (Ewl_Filedialog * fd, const char * path)`

Changes the current path of a filedialog.

Parameters:

fd,: the filedialog to change paths

path,: the new path used for the filedialog

Returns:

Returns no value.

6.25.4.4 `void ewl_filedialog_filter_add (Ewl_Filedialog * fd, const char * name, const char * filter)`

Add the filter named *name* to the combo box in the filedialog.

Parameters:

fd,: The filedialog to work with

name,: The name to display for the filter

filter,: The actual regular expression for the filter

Returns:

Returns no value

6.25.4.5 `int ewl_filedialog_init (Ewl_Filedialog * fd)`

Initialize a new filedialog.

Parameters:

fd,: the filedialog

Returns:

Returns no value.

6.25.4.6 `Ewl_View* ewl_filedialog_list_view_get (Ewl_Filedialog * fd)`

Retrieve the file list view used in this file dialog.

Parameters:

fd,: The filedialog to get the view from

Returns:

Returns the **Ewl_View** (p. 784) set on this file dialog

6.25.4.7 `void ewl_filedialog_list_view_set (Ewl_Filedialog * fd, Ewl_View * view)`

Set the view to be used for displaying the files in the dialog.

Parameters:

fd,: The filedialog to set the view on

view,: The **Ewl_View** (p. 784) to set into the dialog

Returns:

Returns no value.

6.25.4.8 `unsigned int ewl_filedialog_multiselect_get (Ewl_Filedialog * fd)`

gets the multiselect setting of the filedialog

Parameters:

fd,: the filedialog

Returns:

Returns the multi select setting (0|1)

6.25.4.9 Ewl_Widget* ewl_filedialog_multiselect_new (void)

Create a new open filedialog.

Returns:

Returns a new open filedialog if successful, NULL on failure.

6.25.4.10 void ewl_filedialog_multiselect_set (Ewl_Filedialog * *fd*, unsigned int *val*)

Sets the dialog to multiselect or single select.

Parameters:

fd,: the filedialog

val,: 1 to set multiselect, 0 otherwise

Returns:

Returns no value.

6.25.4.11 Ewl_Widget* ewl_filedialog_new (void)

Create a new filedialog.

Returns:

Returns a new filedialog in success, NULL on failure.

6.25.4.12 char* ewl_filedialog_selected_file_get (Ewl_Filedialog * *fd*)

Retrieve the selected filename.

Parameters:

fd,: the filedialog

Returns:

Returns the selected filename including its path

6.25.4.13 void ewl_filedialog_selected_file_set (Ewl_Filedialog * *fd*, const char * *file*)

Set the currently selected file into the file dialog.

Parameters:

fd,: The filedialog to work with

file,: The selected file to set

Returns:

Returns no value

6.25.4.14 `Ecore_List* ewl_filedialog_selected_files_get (Ewl_Filedialog * fd)`

returns all the elements selected by the user

Parameters:

fd,: The filedialog

Returns:

Returns an Ecore_List of selected items

6.25.4.15 `void ewl_filedialog_selected_files_set (Ewl_Filedialog * fd, Ecore_List * files)`

Sets the given files as selected in the filedialog.

Parameters:

fd,: The filedialog to work with

files,: The list of filenames to set selected in the dialog

Returns:

Returns no value

6.25.4.16 `unsigned int ewl_filedialog_show_dot_files_get (Ewl_Filedialog * fd)`

Returns the current dot file setting of the dialog.

Parameters:

fd,: The filedialog to get the value from

Returns:

Returns the show dot file setting of the dialog

6.25.4.17 `void ewl_filedialog_show_dot_files_set (Ewl_Filedialog * fd, unsigned int dot)`

Set if the file dialog should show dot files by default.

Parameters:

fd,: The filedialog to change

dot,: The value to set for show dot files

Returns:

Returns no value.

6.25.4.18 unsigned int ewl_filedialog_show_favorites_get (Ewl_Filedialog * *fd*)

Get the current show favorites setting for the filedialog.

Parameters:

fd,: The filedialog to work with

Returns:

Returns the current show favorites setting for the dialog

6.25.4.19 void ewl_filedialog_show_favorites_set (Ewl_Filedialog * *fd*, unsigned int *show*)

Specify if the favorites column should be shown or not.

Parameters:

fd,: The filedialog to set the value into

show,: The setting for show favorites to set

Returns:

Returns no value.

6.26 Ewl_Filelist: The base widget for the filelists

6.26.1 Detailed Description

A base class to be extended to provide different views on the file list

Remarks:

Inherits from **Ewl_Box** (p. 586).

Data Structures

- struct **Ewl_Filelist**

*Inherits from **Ewl_Box** (p. 586) and extends to display a file list.*

Defines

- #define **EWL_FILELIST**(fl) ((**Ewl_Filelist** *)fl)
- #define **EWL_FILELIST_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FILELIST_TYPE))
- #define **EWL_FILELIST_TYPE** "filelist"

Typedefs

- typedef **Ewl_Filelist** **Ewl_Filelist**

Functions

- void **ewl_filelist_cb_destroy** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_filelist_container_shift_handle** (**Ewl_Filelist** *fl, **Ewl_Container** *c, **Ewl_Widget** *clicked, const char *select_signal, const char *unselect_signal)
- const char * **ewl_filelist_directory_get** (**Ewl_Filelist** *fl)
Retrieves the current directory set on the filelist.
- void **ewl_filelist_directory_read** (**Ewl_Filelist** *fl, const char *dir, unsigned int skip_dot_dot, void(*func)(**Ewl_Filelist** *fl, const char *dir, char *file, void *data), void *data)
- void **ewl_filelist_directory_set** (**Ewl_Filelist** *fl, const char *dir)
Sets the given directory dir as the current directory in the filelist.
- char * **ewl_filelist_expand_path** (**Ewl_Filelist** *fl, const char *dir)
This will attempt to return the full path to the given directory. It should handle things like .. as well.
- const char * **ewl_filelist_filter_get** (**Ewl_Filelist** *fl)
Retrieves the current filter set on the filelist.
- void **ewl_filelist_filter_set** (**Ewl_Filelist** *fl, const char *filter)

Sets the given filter into the filelist.

- `char * ewl_filelist_groupname_get (gid_t st_gid)`
Converts the given group id into a group name.
- `void ewl_filelist_handle_click (Ewl_Filelist *fl, Ewl_Widget *w, Ewl_Event_Mouse_Up *ev, const char *select_state, const char *unselect_state)`
- `Ewl Scrollpane_Flags ewl_filelist_hscroll_flag_get (Ewl_Filelist *fl)`
Retrieves the flags for the horizontal scrollbar.
- `void ewl_filelist_hscroll_flag_set (Ewl_Filelist *fl, Ewl Scrollpane_Flags h)`
Sets the value to use for flags on the horizontal scrollbar.
- `int ewl_filelist_init (Ewl_Filelist *fl)`
Initialzie a filelist to default values.
- `char * ewl_filelist_modtime_get (time_t modtime)`
Converts the given modtime to a human readable string.
- `unsigned int ewl_filelist_multiselect_get (Ewl_Filelist *fl)`
Retrieves the current multiselect state of the filelist.
- `void ewl_filelist_multiselect_set (Ewl_Filelist *fl, unsigned int ms)`
Sets the given multiselect value into the filelist.
- `char * ewl_filelist_perms_get (mode_t st_mode)`
Converts the given mode settings into a human readable string.
- `char * ewl_filelist_selected_file_get (Ewl_Filelist *fl)`
Returns the currently selected file from the filelist.
- `Ewl_Widget * ewl_filelist_selected_file_preview_get (Ewl_Filelist *fl, const char *path)`
Creates and returns a preview widget for the given file.
- `void ewl_filelist_selected_file_set (Ewl_Filelist *fl, const char *file)`
Sets the given file as selected in the filelist.
- `void ewl_filelist_selected_files_change_notify (Ewl_Filelist *fl)`
Notifies interested consumers that the filelist has changed selected values.
- `Ecore_List * ewl_filelist_selected_files_get (Ewl_Filelist *fl)`
Retrieves the list of selected files in the filelist.
- `void ewl_filelist_selected_files_set (Ewl_Filelist *fl, Ecore_List *files)`
Sets the given files as selected in the filelist.
- `void ewl_filelist_selected_signal_all (Ewl_Filelist *fl, const char *signal)`
Signals all of the selected widgets with the given signal.

- unsigned int **ewl_filelist_show_dot_files_get** (**Ewl_Filelist** *fl)
Retrieves the current show dot files setting for the filelist.
- void **ewl_filelist_show_dot_files_set** (**Ewl_Filelist** *fl, unsigned int dot)
Sets the show dot files setting to the given value.
- char * **ewl_filelist_size_get** (off_t st_size)
Converts the given size into a human readable format.
- const char * **ewl_filelist_stock_icon_get** (**Ewl_Filelist** *fl, const char *path)
Retrieves the stock icon for the given file.
- char * **ewl_filelist_username_get** (uid_t st_uid)
Convertes the given user id into the appropriate user name.
- **Ewl Scrollpane_Flags** **ewl_filelist_vscroll_flag_get** (**Ewl_Filelist** *fl)
Retrieves the flags for the vertical scrollbar.
- void **ewl_filelist_vscroll_flag_set** (**Ewl_Filelist** *fl, **Ewl Scrollpane_Flags** v)
Sets the value to use for flags on the vertical scrollbar.

6.26.2 Define Documentation

6.26.2.1 #define EWL_FILELIST(fl) ((Ewl_Filelist *)fl)

Typecasts a pointer to an **Ewl_Filelist** (p. 658) pointer.

6.26.2.2 #define EWL_FILELIST_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FILELIST_TYPE))

Returns TRUE if the widget is an **Ewl_Filelist** (p. 658), FALSE otherwise

6.26.2.3 #define EWL_FILELIST_TYPE "filelist"

The type name for the **Ewl_Filelist** (p. 658) widget

6.26.3 Typedef Documentation

6.26.3.1 typedef struct Ewl_Filelist Ewl_Filelist

The **Ewl_Filelist** (p. 658) provides a base class for the file lists

6.26.4 Function Documentation

6.26.4.1 void ewl_filelist_cb_destroy (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.26.4.2 void ewl_filelist_container_shift_handle (Ewl_Filelist * *fl*,
Ewl_Container * *c*, Ewl_Widget * *clicked*, const char * *select_signal*,
const char * *unselect_signal*)

6.26.4.3 const char* ewl_filelist_directory_get (Ewl_Filelist * *fl*)

Retrieves the current directory set on the filelist.

Parameters:

fl,: The filelist to get the current directory from

Returns:

Returns the current directory

6.26.4.4 void ewl_filelist_directory_read (Ewl_Filelist * *fl*, const char * *dir*,
unsigned int *skip_dot_dot*, void(*) (Ewl_Filelist **fl*, const char **dir*, char
**file*, void **data*) *func*, void * *data*)

6.26.4.5 void ewl_filelist_directory_set (Ewl_Filelist * *fl*, const char * *dir*)

Sets the given directory *dir* as the current directory in the filelist.

Parameters:

fl,: The filelist to set the directory into

dir,: The directory to set

Returns:

Returns no value

6.26.4.6 char* ewl_filelist_expand_path (Ewl_Filelist * *fl*, const char * *dir*)

This will attempt to return the full path to the given directory. It should handle things like .. as well.

Parameters:

fl,: The filelist to get the path from

dir,: The dir name to append to the path

Returns:

Returns the full path to the given directory

6.26.4.7 `const char* ewl_filelist_filter_get (Ewl_Filelist * fl)`

Retrieves the current filter set on the filelist.

Parameters:

fl,: The filelist to get the filter from

Returns:

Returns the current filter

6.26.4.8 `void ewl_filelist_filter_set (Ewl_Filelist * fl, const char * filter)`

Sets the given filter into the filelist.

Parameters:

fl,: The filelist to set the filter into

filter,: The filter to set

Returns:

Returns no value.

6.26.4.9 `char* ewl_filelist_groupname_get (gid_t st_gid)`

Converts the given group id into a group name.

Parameters:

st_gid,: The group id to convert

Returns:

Returns the group name for the given id

6.26.4.10 `void ewl_filelist_handle_click (Ewl_Filelist * fl, Ewl_Widget * w,
Ewl_Event_Mouse_Up * ev, const char * select_state, const char *
unselect_state)`**6.26.4.11** `Ewl_Scrollpane_Flags ewl_filelist_hscroll_flag_get (Ewl_Filelist * fl)`

Retrieves the flags for the horizontal scrollbar.

Parameters:

fl,: The filelist to work with

Returns:

Returns the flags for the horizontal scrollbar

6.26.4.12 void ewl_filelist_hscroll_flag_set (Ewl_Filelist * *fl*,
Ewl_Scrollpane_Flags *h*)

Sets the value to use for flags on the horizontal scrollbar.

Parameters:

fl,: The filelist to work with

h,: The value to set for the horizontal scrollbar

Returns:

Returns no value

6.26.4.13 int ewl_filelist_init (Ewl_Filelist * *fl*)

Initialzie a filelist to default values.

Parameters:

fl,: The filelist to initialize

Returns:

Returns TRUE on success or FALSE on failure

6.26.4.14 char* ewl_filelist_modtime_get (time_t *st_modtime*)

Converts the given modtime to a human readable string.

Parameters:

st_modtime,: The modification time to convert

Returns:

Returns the string version of the modtime

6.26.4.15 unsigned int ewl_filelist_multiselect_get (Ewl_Filelist * *fl*)

Retrieves the current multiselect state of the filelist.

Parameters:

fl,: The filelist to get the multiselect value from

Returns:

Returns the current multiselect state of the filelist

6.26.4.16 `void ewl_filelist_multiselect_set (Ewl_Filelist * fl, unsigned int ms)`

Sets the given multiselect value into the filelist.

Parameters:

fl,: The filelist to set the multiselect value into
ms,: The multiselect value to set

Returns:

Returns no value

6.26.4.17 `char* ewl_filelist_perms_get (mode_t st_mode)`

Converts the given mode settings into a human readable string.

Parameters:

st_mode,: The mode setting to convert

Returns:

Returns the string of the given mode setting

6.26.4.18 `char* ewl_filelist_selected_file_get (Ewl_Filelist * fl)`

Returns the currently selected file from the filelist.

Parameters:

fl,: The filelist to get the selected file from

Returns:

Returns the file currently selected in the filelist

6.26.4.19 `Ewl_Widget* ewl_filelist_selected_file_preview_get (Ewl_Filelist * fl,
const char * path)`

Creates and returns a preview widget for the given file.

Parameters:

fl,: The `Ewl_Filelist` (p. 658) to work with
path,: The file to get the preview for

Returns:

Returns the preview widget for the given file

6.26.4.20 void ewl_filelist_selected_file_set (Ewl_Filelist * *fl*, const char * *file*)

Sets the given file as selected in the filelist.

Parameters:

fl,: The filelist to set the selected file into

file,: The file to set selected

Returns:

Returns no value

6.26.4.21 void ewl_filelist_selected_files_change_notify (Ewl_Filelist * *fl*)

Notifies interested consumers that the filelist has changed selected values.

Parameters:

fl,: The filelist to work with

Returns:

Returns no value.

6.26.4.22 Ecore_List* ewl_filelist_selected_files_get (Ewl_Filelist * *fl*)

Retrieves the list of selected files in the filelist.

Parameters:

fl,: The filelist to get the selected files from

Returns:

Returns the Ecore_List of selected files in the filelist

6.26.4.23 void ewl_filelist_selected_files_set (Ewl_Filelist * *fl*, Ecore_List * *files*)

Sets the given files as selected in the filelist.

Parameters:

fl,: The filelist to set the selected files into

files,: The Ecore_List of files to set

Returns:

Returns no value.

6.26.4.24 `void ewl_filelist_selected_signal_all (Ewl_Filelist * fl, const char * signal)`

Signals all of the selected widgets with the given signal.

Parameters:

fl,: The filelist to work with

signal,: The signal to send

Returns:

Returns no value

6.26.4.25 `unsigned int ewl_filelist_show_dot_files_get (Ewl_Filelist * fl)`

Retrieves the current show dot files setting for the filelist.

Parameters:

fl,: The filelist to get the value from

Returns:

Returns the current show dot files setting of the filelist

6.26.4.26 `void ewl_filelist_show_dot_files_set (Ewl_Filelist * fl, unsigned int dot)`

Sets the show dot files setting to the given value.

Parameters:

fl,: The filelist to set the show dot files setting into

dot,: The value to set into the show dot files field

Returns:

Returns no value.

6.26.4.27 `char* ewl_filelist_size_get (off_t st_size)`

Converts the given size into a human readable format.

Parameters:

st_size,: The size to convert

Returns:

Returns a string representation of the given size

6.26.4.28 `const char* ewl_filelist_stock_icon_get (Ewl_Filelist * fl, const char * path)`

Retrieves the stock icon for the given file.

Parameters:

fl,: The filelist to work with

path,: The path to get the icon for

Returns:

Returns the stock icon for the given file

6.26.4.29 `char* ewl_filelist_username_get (uid_t st_uid)`

Convertes the given user id into the appropriate user name.

Parameters:

st_uid,: The userid to lookup

Returns:

Returns the user name for the given user id

6.26.4.30 `Ewl_Scrollpane_Flags ewl_filelist_vscroll_flag_get (Ewl_Filelist * fl)`

Retrieves the flags for the vertical scrollbar.

Parameters:

fl,: The filelist to work with

Returns:

Returns the flags for the vertical scrollbar

6.26.4.31 `void ewl_filelist_vscroll_flag_set (Ewl_Filelist * fl,
Ewl_Scrollpane_Flags v)`

Sets the value to use for flags on the vertical scrollbar.

Parameters:

fl,: The filelist to work with

v,: The value to set for the vertical scrollbar

Returns:

Returns no value

6.27 Ewl_Filelist_Column: An expanding column view

6.27.1 Detailed Description

A view of the files using a new column for each directory.

Remarks:

Inherits from **Ewl_Filelist** (p. 658).

Data Structures

- struct **Ewl_Filelist_Column**

*Inherits from **Ewl_Filelist** (p. 658) and extends to provide a column view.*

Defines

- #define **EWL_FILELIST_COLUMN**(fl) ((**Ewl_Filelist_Column** *)fl)
- #define **EWL_FILELIST_COLUMN_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_FILELIST_COLUMN_TYPE**))
- #define **EWL_FILELIST_COLUMN_TYPE** "filelist_column"

Typedefs

- typedef **Ewl_Filelist_Column** **Ewl_Filelist_Column**

Functions

- void **ewl_filelist_column_dir_change** (**Ewl_Filelist** *fl)
The callback for a directory change.
- const char * **ewl_filelist_column_filename_get** (**Ewl_Filelist** *fl, void *item)
Retrieves the filename for the given item.
- int **ewl_filelist_column_init** (**Ewl_Filelist_Column** *fl)
*Initializes an **Ewl_Filelist_Column** (p. 661) to default values.*
- **Ewl_Widget** * **ewl_filelist_column_new** (void)
*Creates a new **Ewl_Filelist_Column** (p. 661) widget.*
- void **ewl_filelist_column_selected_file_add** (**Ewl_Filelist** *fl, const char *file)
The callback to notify of selected files changing.
- void **ewl_filelist_column_selected_unselect** (**Ewl_Filelist** *fl)
- void **ewl_filelist_column_shift_handle** (**Ewl_Filelist** *fl, **Ewl_Widget** *clicked)
- **Ewl_View** * **ewl_filelist_column_view_get** (void)
*Retrieves the needed **Ewl_View** (p. 784) to use the column view.*

6.27.2 Define Documentation

6.27.2.1 `#define EWL_FILELIST_COLUMN(fl) ((Ewl_Filelist_Column *)fl)`

Typecasts a pointer to an `Ewl_Filelist_Column` (p. 661) pointer

6.27.2.2 `#define EWL_FILELIST_COLUMN_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FILELIST_COLUMN_TYPE))`

Returns TRUE if the widget is an `Ewl_Filelist_Column` (p. 661), FALSE otherwise

6.27.2.3 `#define EWL_FILELIST_COLUMN_TYPE "filelist_column"`

The type name for the `Ewl_Filelist_Column` (p. 661) widget

6.27.3 Typedef Documentation

6.27.3.1 `typedef struct Ewl_Filelist_Column Ewl_Filelist_Column`

The `Ewl_Filelist_Column` (p. 661) provides a column view of the files

6.27.4 Function Documentation

6.27.4.1 `void ewl_filelist_column_dir_change (Ewl_Filelist * fl)`

The callback for a directory change.

Parameters:

fl,: The filelist to work with

Returns:

Returns no value

6.27.4.2 `const char* ewl_filelist_column_filename_get (Ewl_Filelist * fl, void * item)`

Retrieves the filename for the given item.

Parameters:

fl,: The filelist to work with

item,: The item to get the name from

Returns:

Returns the filename for the given item

6.27.4.3 int ewl_filelist_column_init (Ewl_Filelist_Column * *fl*)

Initializes an **Ewl_Filelist_Column** (p. 661) to default values.

Parameters:

fl,: The filelist to work with

Returns:

Returns TRUE on success or FALSE on failure

6.27.4.4 Ewl_Widget* ewl_filelist_column_new (void)

Creates a new **Ewl_Filelist_Column** (p. 661) widget.

Returns:

Returns a new **Ewl_Filelist_Column** (p. 661) or NULL on failure

6.27.4.5 void ewl_filelist_column_selected_file_add (Ewl_Filelist * *fl*, const char * *file*)

The callback to notify of selected files changing.

Parameters:

fl,: The filelist to work with

file,: The file to set selected

Returns:

Returns no value

6.27.4.6 void ewl_filelist_column_selected_unselect (Ewl_Filelist * *fl*)**6.27.4.7 void ewl_filelist_column_shift_handle (Ewl_Filelist * *fl*, Ewl_Widget * *clicked*)****6.27.4.8 Ewl_View* ewl_filelist_column_view_get (void)**

Retrieves the needed **Ewl_View** (p. 784) to use the column view.

Returns:

Returns the **Ewl_View** (p. 784) needed to display the filelist_column

6.28 Ewl_Filelist_Icon: A icon file view

6.28.1 Detailed Description

A view of the files using simple name/icon graphics

Remarks:

Inherits from **Ewl_Filelist** (p. 658).

Data Structures

- struct **Ewl_Filelist_Icon**
*Inherits from **Ewl_Filelist** (p. 658) and extends for an icon view.*

Defines

- #define **EWL_FILELIST_ICON**(fl) ((**Ewl_Filelist_Icon** *)fl)
- #define **EWL_FILELIST_ICON_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_FILELIST_ICON_TYPE**))
- #define **EWL_FILELIST_ICON_TYPE** "filelist_icon"

Typedefs

- typedef **Ewl_Filelist_Icon** **Ewl_Filelist_Icon**

Functions

- void **ewl_filelist_icon_dir_change** (**Ewl_Filelist** *fl)
The callback for a directory change.
- const char * **ewl_filelist_icon_filename_get** (**Ewl_Filelist** *fl, void *item)
Retrieves the filename for the given item.
- int **ewl_filelist_icon_init** (**Ewl_Filelist_Icon** *fl)
*Initializes an **Ewl_Filelist_Icon** (p. 664) to default values.*
- **Ewl_Widget** * **ewl_filelist_icon_new** (void)
*Creates a new **Ewl_Filelist_Icon** (p. 664) widget.*
- void **ewl_filelist_icon_selected_file_add** (**Ewl_Filelist** *fl, const char *file)
The callback to notify of selected files changing.
- void **ewl_filelist_icon_selected_unselect** (**Ewl_Filelist** *fl)
- void **ewl_filelist_icon_shift_handle** (**Ewl_Filelist** *fl, **Ewl_Widget** *clicked)
- **Ewl_View** * **ewl_filelist_icon_view_get** (void)
*Retrieves the needed **Ewl_View** (p. 784) to use the icon view.*

6.28.2 Define Documentation

6.28.2.1 `#define EWL_FILELIST_ICON(fl) ((Ewl_Filelist_Icon *)fl)`

Typecasts a pointer to an **Ewl_Filelist_Icon** (p. 664) pointer

6.28.2.2 `#define EWL_FILELIST_ICON_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FILELIST_ICON_TYPE))`

Returns TRUE if the widget is an **Ewl_Filelist_Icon** (p. 664), FALSE otherwise

6.28.2.3 `#define EWL_FILELIST_ICON_TYPE "filelist_icon"`

The type name for the **Ewl_Filelist_Icon** (p. 664) widget

6.28.3 Typedef Documentation

6.28.3.1 `typedef struct Ewl_Filelist_Icon Ewl_Filelist_Icon`

The **Ewl_Filelist_Icon** (p. 664) provides an icon view of the files

6.28.4 Function Documentation

6.28.4.1 `void ewl_filelist_icon_dir_change (Ewl_Filelist * fl)`

The callback for a directory change.

Parameters:

fl,: The filelist to work with

Returns:

Returns no value

6.28.4.2 `const char* ewl_filelist_icon_filename_get (Ewl_Filelist * fl, void * item)`

Retrieves the filename for the given item.

Parameters:

fl,: The filelist to work with

item,: The item to get the name from

Returns:

Returns the filename for the given item

6.28.4.3 int ewl_filelist_icon_init (Ewl_Filelist_Icon * *fl*)

Initializes an **Ewl_Filelist_Icon** (p. 664) to default values.

Parameters:

fl,: The filelist to work with

Returns:

Returns TRUE on success or FALSE on failure

6.28.4.4 Ewl_Widget* ewl_filelist_icon_new (void)

Creates a new **Ewl_Filelist_Icon** (p. 664) widget.

Returns:

Returns a new **Ewl_Filelist_Icon** (p. 664) or NULL on failure

6.28.4.5 void ewl_filelist_icon_selected_file_add (Ewl_Filelist * *fl*, const char * *file*)

The callback to notify of selected files changing.

Parameters:

fl,: The filelist to work with

file,: The file to set selected

Returns:

Returns no value

6.28.4.6 void ewl_filelist_icon_selected_unselect (Ewl_Filelist * *fl*)**6.28.4.7 void ewl_filelist_icon_shift_handle (Ewl_Filelist * *fl*, Ewl_Widget * *clicked*)****6.28.4.8 Ewl_View* ewl_filelist_icon_view_get (void)**

Retrieves the needed **Ewl_View** (p. 784) to use the icon view.

Returns:

Returns the **Ewl_View** (p. 784) needed to display the filelist_icon

6.29 Ewl_Filelist_List: A listview of the files

6.29.1 Detailed Description

Provides a list view of the files giving name, size, etc information

Remarks:

Inherits from **Ewl_Filelist** (p. 658).

Data Structures

- struct **Ewl_Filelist_List**

*Inherits from **Ewl_Filelist** (p. 658) and extends to create a list view.*

Defines

- #define **EWL_FILELIST_LIST**(fl) ((**Ewl_Filelist_List** *)fl)
- #define **EWL_FILELIST_LIST_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_FILELIST_LIST_TYPE**))
- #define **EWL_FILELIST_LIST_TYPE** "filelist_list"

Typedefs

- typedef **Ewl_Filelist_List** **Ewl_Filelist_List**

Functions

- void **ewl_filelist_list_dir_change** (**Ewl_Filelist** *fl)
The callback to notify of a directory change.
- const char * **ewl_filelist_list_filename_get** (**Ewl_Filelist** *fl, void *item)
Retrieves the filename for the given item.
- int **ewl_filelist_list_init** (**Ewl_Filelist_List** *fl)
*Initializes an **Ewl_Filelist_List** (p. 665) widget to default values.*
- **Ewl_Widget** * **ewl_filelist_list_new** (void)
*Creates a new **Ewl_Filelist_List** (p. 665) widget.*
- void **ewl_filelist_list_selected_file_add** (**Ewl_Filelist** *fl, const char *file)
- void **ewl_filelist_list_selected_unselect** (**Ewl_Filelist** *fl)
- void **ewl_filelist_list_shift_handle** (**Ewl_Filelist** *fl, **Ewl_Widget** *clicked)
- **Ewl_View** * **ewl_filelist_list_view_get** (void)
*Retrieves the **Ewl_View** (p. 784) needed to use the filelist list view.*

6.29.2 Define Documentation

6.29.2.1 `#define EWL_FILELIST_LIST(fl) ((Ewl_Filelist_List *)fl)`

Typecasts a pointer to an **Ewl_Filelist_List** (p. 665) pointer.

6.29.2.2 `#define EWL_FILELIST_LIST_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FILELIST_LIST_TYPE))`

Returns TRUE if the widget is an **Ewl_Filelist_List** (p. 665), FALSE otherwise

6.29.2.3 `#define EWL_FILELIST_LIST_TYPE "filelist_list"`

The type name for the **Ewl_Filelist_List** (p. 665) widget

6.29.3 Typedef Documentation

6.29.3.1 `typedef struct Ewl_Filelist_List Ewl_Filelist_List`

The **Ewl_Filelist_List** (p. 665) widget provides a list view of the files

6.29.4 Function Documentation

6.29.4.1 `void ewl_filelist_list_dir_change (Ewl_Filelist * fl)`

The callback to notify of a directory change.

Parameters:

fl,: The filelist to work with

Returns:

Returns no value

6.29.4.2 `const char* ewl_filelist_list_filename_get (Ewl_Filelist * fl, void * item)`

Retrieves the filename for the given item.

Parameters:

fl,: The filelist to work with

item,: The item to get the name from

Returns:

Returns the filename for the given item

6.29.4.3 `int ewl_filelist_list_init (Ewl_Filelist_List * fl)`

Initializes an **Ewl_Filelist_List** (p. 665) widget to default values.

Parameters:

fl,: The **Ewl_Filelist_List** (p. 665) to initialize

Returns:

Returns TRUE on success or FALSE on failure

6.29.4.4 `Ewl_Widget* ewl_filelist_list_new (void)`

Creates a new **Ewl_Filelist_List** (p. 665) widget.

Returns:

Returns a new **Ewl_Filelist_List** (p. 665) widget or NULL on failure

6.29.4.5 `void ewl_filelist_list_selected_file_add (Ewl_Filelist * fl, const char * file)`**6.29.4.6** `void ewl_filelist_list_selected_unselect (Ewl_Filelist * fl)`**6.29.4.7** `void ewl_filelist_list_shift_handle (Ewl_Filelist * fl, Ewl_Widget * clicked)`**6.29.4.8** `Ewl_View* ewl_filelist_list_view_get (void)`

Retrieves the **Ewl_View** (p. 784) needed to use the filelist list view.

Returns:

Returns the view for the filelist list

6.30 Ewl_Filelist_Tree: A treeview of the files

6.30.1 Detailed Description

Provides a tree view of the files giving name, size, etc information

Remarks:

Inherits from **Ewl_Filelist** (p. 658).

Data Structures

- struct **Ewl_Filelist_Tree**

*Inherits from **Ewl_Filelist** (p. 658) and extends to provide a tree view.*

Defines

- `#define EWL_FILELIST_TREE(fl) ((Ewl_Filelist_Tree *)fl)`
- `#define EWL_FILELIST_TREE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FILELIST_TREE_TYPE))`
- `#define EWL_FILELIST_TREE_TYPE "filelist_tree"`

Typedefs

- `typedef Ewl_Filelist_Tree Ewl_Filelist_Tree`

Functions

- `void ewl_filelist_tree_dir_change (Ewl_Filelist *fl)`
The callback to notify of a directory change.
- `const char * ewl_filelist_tree_filename_get (Ewl_Filelist *fl, void *item)`
Retrieves the filename for the given item.
- `int ewl_filelist_tree_init (Ewl_Filelist_Tree *fl)`
*Initializes an **Ewl_Filelist_Tree** (p. 666) widget to default values.*
- `Ewl_Widget * ewl_filelist_tree_new (void)`
*Creates a new **Ewl_Filelist_Tree** (p. 666) widget.*
- `void ewl_filelist_tree_selected_file_add (Ewl_Filelist *fl, const char *file)`
- `void ewl_filelist_tree_selected_unselect (Ewl_Filelist *fl)`
- `void ewl_filelist_tree_shift_handle (Ewl_Filelist *fl, Ewl_Widget *clicked)`
- `Ewl_View * ewl_filelist_tree_view_get (void)`
*Retrieves the **Ewl_View** (p. 784) needed to use the filelist tree view.*

6.30.2 Define Documentation

6.30.2.1 `#define EWL_FILELIST_TREE(fl) ((Ewl_Filelist_Tree *)fl)`

Typecasts a pointer to an `Ewl_Filelist_Tree` (p. 666) pointer.

6.30.2.2 `#define EWL_FILELIST_TREE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FILELIST_TREE_TYPE))`

Returns TRUE if the widget is an `Ewl_Filelist_Tree` (p. 666), FALSE otherwise

6.30.2.3 `#define EWL_FILELIST_TREE_TYPE "filelist_tree"`

The type name for the `Ewl_Filelist_Tree` (p. 666) widget

6.30.3 Typedef Documentation

6.30.3.1 `typedef struct Ewl_Filelist_Tree Ewl_Filelist_Tree`

The `Ewl_Filelist_Tree` (p. 666) widget provides a tree view of the files

6.30.4 Function Documentation

6.30.4.1 `void ewl_filelist_tree_dir_change (Ewl_Filelist * fl)`

The callback to notify of a directory change.

Parameters:

fl,: The filelist to work with

Returns:

Returns no value

6.30.4.2 `const char* ewl_filelist_tree_filename_get (Ewl_Filelist * fl, void * item)`

Retrieves the filename for the given item.

Parameters:

fl,: The filelist to work with

item,: The item to get the name from

Returns:

Returns the filename for the given item

6.30.4.3 int ewl_filelist_tree_init (Ewl_Filelist_Tree * *fl*)

Initializes an **Ewl_Filelist_Tree** (p. 666) widget to default values.

Parameters:

fl: The **Ewl_Filelist_Tree** (p. 666) to initialize

Returns:

Returns TRUE on success or FALSE on failure

6.30.4.4 Ewl_Widget* ewl_filelist_tree_new (void)

Creates a new **Ewl_Filelist_Tree** (p. 666) widget.

Returns:

Returns a new **Ewl_Filelist_Tree** (p. 666) widget or NULL on failure

6.30.4.5 void ewl_filelist_tree_selected_file_add (Ewl_Filelist * *fl*, const char * *file*)**6.30.4.6 void ewl_filelist_tree_selected_unselect (Ewl_Filelist * *fl*)****6.30.4.7 void ewl_filelist_tree_shift_handle (Ewl_Filelist * *fl*, Ewl_Widget * *clicked*)****6.30.4.8 Ewl_View* ewl_filelist_tree_view_get (void)**

Retrieves the **Ewl_View** (p. 784) needed to use the filelist tree view.

Returns:

Returns the view for the filelist tree

6.31 Ewl_Filepicker: Allows selecting files from a list

6.31.1 Detailed Description

A widget to allow selecting ifles

Remarks:

Inherits from **Ewl_Box** (p. 586).

6.31.2 Tutorial

The **Ewl_Filepicker** (p. 668) widget provides a method for selecting files. The filepicker can support multiple views of the files, this also allows the app to provide it's own view.

```
Ewl_Widget *fp;  
fp = ewl_filepicker_new();  
ewl_widget_show(fp);
```

The filepicker will default to the users home directory. If that dosen't exist it will attemp to use /tmp, failing that it will fall back to the / directory. The following two calls can be used to retrieve or set the current directory.

```
ewl_filepicker_directory_set(fp);  
ewl_filepicker_directory_get(fp);
```

Along with changing the directory you can also specify if the picker will allow for multiple file selections, showing dot files, showing the favorites panel or setting the file filter. This is done with the following code:

```
ewl_filepicker_multiselect_set(fp, TRUE);  
ms = ewl_filepicker_multiselect_get(fp);  
  
ewl_filepicker_show_dot_files_set(fp, TRUE);  
dot = ewl_filepicker_show_dot_files_get(fp);  
  
ewl_filepicker_show_favorites_set(fp, TRUE);  
fav = ewl_filepicker_show_favorites_get(fp);  
  
ewl_filepicker_filter_set(fp, filter);  
filter = ewl_filepicker_filter_get(fp);
```

You can also change the view used for the file list.

```
ewl_filepicker_list_view_set(fp, view);  
view = ewl_filepicker_list_view_get(fp);
```

6.31.3 Adding Views

A view of the file list in the filepicker is a subclass of the **Ewl_Filelist** (p. 658) widget. These subclasses will provide the code for the actual list display. They will receive callbacks as settings are changed on the filepicker to notify of the display changes.

If you wish to write your own view you will need to create an **Ewl_View** (p. 784) and set the constructor for your widget as the construct field. The constructor will need to setup the appropriate callbacks to as specified in the **Ewl_Filelist** (p. 658) structure.

Data Structures

- struct **Ewl_Filepicker**

*Inherits from **Ewl_Box** (p. 586) and extends to provide a filepicker.*

Defines

- #define **EWL_FILEPICKER**(fp) ((**Ewl_Filepicker** *)fp)
- #define **EWL_FILEPICKER_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_FILEPICKER_TYPE**))
- #define **EWL_FILEPICKER_TYPE** "filepicker"

Typedefs

- typedef **Ewl_Filepicker** **Ewl_Filepicker**

Functions

- const char * **ewl_filepicker_directory_get** (**Ewl_Filepicker** *fp)
Retrieve the current directory from the filepicker.
- void **ewl_filepicker_directory_set** (**Ewl_Filepicker** *fp, const char *dir)
Set the current directory to be displayed in the filepicker.
- void **ewl_filepicker_filter_add** (**Ewl_Filepicker** *fp, const char *name, const char *filter)
Add the filter named name to the combo box in the filepicker.
- const char * **ewl_filepicker_filter_get** (**Ewl_Filepicker** *fp)
Retrieves the current filter used in the filepicker.
- void **ewl_filepicker_filter_set** (**Ewl_Filepicker** *fp, const char *filter)
Set the given filter into the file picker.
- int **ewl_filepicker_init** (**Ewl_Filepicker** *fp)
*Initialize an **Ewl_Filepicker** (p. 668) to default values.*
- **Ewl_View** * **ewl_filepicker_list_view_get** (**Ewl_Filepicker** *fp)
Retrieves the current view used for the file list.
- void **ewl_filepicker_list_view_set** (**Ewl_Filepicker** *fp, **Ewl_View** *view)
Sets the given view to be used to display the file list.
- unsigned int **ewl_filepicker_multiselect_get** (**Ewl_Filepicker** *fp)
Retrieves the current multi-select value from the filepicker.
- void **ewl_filepicker_multiselect_set** (**Ewl_Filepicker** *fp, unsigned int ms)
Sets the multiselect state of the filepicker to the given value.

- **Ewl_Widget * ewl_filepicker_new** (void)
*Creates a new **Ewl_Filepicker** (p. 668) widget.*
- **char * ewl_filepicker_selected_file_get** (Ewl_Filepicker *fp)
Retrieves the currently selected file from the filepicker.
- **void ewl_filepicker_selected_file_set** (Ewl_Filepicker *fp, const char *file)
Sets the given file as selected in the filepicker.
- **Ecore_List * ewl_filepicker_selected_files_get** (Ewl_Filepicker *fp)
Retrieves the list of selected files in the filepicker.
- **void ewl_filepicker_selected_files_set** (Ewl_Filepicker *fp, Ecore_List *files)
Sets the given files as selected in the filepicker.
- **unsigned int ewl_filepicker_show_dot_files_get** (Ewl_Filepicker *fp)
Retrieves the current show dot files setting of the filepicker.
- **void ewl_filepicker_show_dot_files_set** (Ewl_Filepicker *fp, unsigned int dot)
Set the show dot files setting of the filepicker to the given value.
- **unsigned int ewl_filepicker_show_favorites_get** (Ewl_Filepicker *fp)
Retrieves the current show favorites setting of the filepicker.
- **void ewl_filepicker_show_favorites_set** (Ewl_Filepicker *fp, unsigned int show)
Sets the show favorites setting of the file picker.

6.31.4 Define Documentation

6.31.4.1 **#define EWL_FILEPICKER(fp) ((Ewl_Filepicker *)fp)**

Typecast a pointer to an **Ewl_Filepicker** (p. 668) pointer.

6.31.4.2 **#define EWL_FILEPICKER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FILEPICKER_TYPE))**

Returns TRUE if the widget is an **Ewl_Filepicker** (p. 668), FALSE otherwise

6.31.4.3 **#define EWL_FILEPICKER_TYPE "filepicker"**

The type name for the **Ewl_Filepicker** (p. 668) widget

6.31.5 Typedef Documentation

6.31.5.1 **typedef struct Ewl_Filepicker Ewl_Filepicker**

The **Ewl_Filepicker** (p. 668) provides a widget for picking files

6.31.6 Function Documentation

6.31.6.1 `const char* ewl_filepicker_directory_get (Ewl_Filepicker * fp)`

Retrieve the current directory from the filepicker.

Parameters:

fp,: The filepicker to get the directory from

Returns:

Returns the current directory displayed in the file dialog

6.31.6.2 `void ewl_filepicker_directory_set (Ewl_Filepicker * fp, const char * dir)`

Set the current directory to be displayed in the filepicker.

Parameters:

fp,: The filepicker to set the directory on

dir,: The directory to set

Returns:

Returns no value.

6.31.6.3 `void ewl_filepicker_filter_add (Ewl_Filepicker * fp, const char * name, const char * filter)`

Add the filter named *name* to the combo box in the filepicker.

Parameters:

fp,: The filepicker to work with

name,: The name to display for the filter

filter,: The actual regular expression for the filter

Returns:

Returns no value

6.31.6.4 `const char* ewl_filepicker_filter_get (Ewl_Filepicker * fp)`

Retrieves the current filter used in the filepicker.

Parameters:

fp,: The filepicker to get the filter from

Returns:

Returns the current filter used in the filepicker

6.31.6.5 void ewl_filepicker_filter_set (Ewl_Filepicker * *fp*, const char * *filter*)

Set the given filter into the file picker.

Parameters:

fp,: The filepicker to set the filter on

filter,: The filter to set on the filepicker

Returns:

Returns no value.

6.31.6.6 int ewl_filepicker_init (Ewl_Filepicker * *fp*)

Initialize an **Ewl_Filepicker** (p. 668) to default values.

Parameters:

fp,: The filepicker to initialize

Returns:

Returns TRUE on success or FALSE on failure

6.31.6.7 Ewl_View* ewl_filepicker_list_view_get (Ewl_Filepicker * *fp*)

Retrieves the current view used for the file list.

Parameters:

fp,: The filepicker to get the value from

Returns:

Returns the current view set on the filepicker

6.31.6.8 void ewl_filepicker_list_view_set (Ewl_Filepicker * *fp*, Ewl_View * *view*)

Sets the given view to be used to display the file list.

Parameters:

fp,: The filepicker to change

view,: The view to use for the file list

Returns:

Returns no value

6.31.6.9 unsigned int ewl_filepicker_multiselect_get (Ewl_Filepicker * *fp*)

Retrieves the current multi-select value from the filepicker.

Parameters:

fp,: The filepicker to get the multiselect value from

Returns:

Returns the current multi-select value of the filepicker

6.31.6.10 void ewl_filepicker_multiselect_set (Ewl_Filepicker * *fp*, unsigned int *ms*)

Sets the multiselect state of the filepicker to the given value.

Parameters:

fp,: The filepicker to set the multi-select flag

ms,: The multi-select setting to place on the filepicker

Returns:

Returns no value.

6.31.6.11 Ewl_Widget* ewl_filepicker_new (void)

Creates a new **Ewl_Filepicker** (p. 668) widget.

Returns:

Returns a new **Ewl_Filepicker** (p. 668) widget or NULL on failure

6.31.6.12 char* ewl_filepicker_selected_file_get (Ewl_Filepicker * *fp*)

Retrieves the currently selected file from the filepicker.

Parameters:

fp,: The filepicker to get the selected file from

Returns:

Returns the currently selected file in the filepicker

6.31.6.13 `void ewl_filepicker_selected_file_set (Ewl_Filepicker * fp, const char * file)`

Sets the given file as selected in the filepicker.

Parameters:

fp,: The filepicker to set the selected file into

file,: The file to set selected in the filepicker

Returns:

Returns no value

6.31.6.14 `Ecore_List* ewl_filepicker_selected_files_get (Ewl_Filepicker * fp)`

Retrieves the list of selected files in the filepicker.

Parameters:

fp,: The filepicker to work with

Returns:

Returns the Ecore_List of selected files in the filepicker

6.31.6.15 `void ewl_filepicker_selected_files_set (Ewl_Filepicker * fp, Ecore_List * files)`

Sets the given files as selected in the filepicker.

Parameters:

fp,: The filepicker to set the selected files into

files,: The Ecore_List of selected files to set into the filepicker

Returns:

Returns no value

6.31.6.16 `unsigned int ewl_filepicker_show_dot_files_get (Ewl_Filepicker * fp)`

Retrieves the current show dot files setting of the filepicker.

Parameters:

fp,: The filepicker to get the showdot files setting from

Returns:

Returns the current show dot files setting of the filepicker

6.31.6.17 void ewl_filepicker_show_dot_files_set (Ewl_Filepicker * *fp*, unsigned int *dot*)

Set the show dot files setting of the filepicker to the given value.

Parameters:

fp,: The filepicker to set the value into

dot,: The show dot files setting to set.

Returns:

Returns no value.

6.31.6.18 unsigned int ewl_filepicker_show_favorites_get (Ewl_Filepicker * *fp*)

Retrieves the current show favorites setting of the filepicker.

Parameters:

fp,: The filepicker to get the show favorites value from

Returns:

Returns the current show favorites setting of the filepicker

6.31.6.19 void ewl_filepicker_show_favorites_set (Ewl_Filepicker * *fp*, unsigned int *show*)

Sets the show favorites setting of the file picker.

Parameters:

fp,: The filepicker to change

show,: The show favorites setting to apply

Returns:

Returns no value

6.32 Ewl_Floater: A Floating Container

6.32.1 Detailed Description

Defines a widget for layering above other widgets in EWL's drawing area, with the ability to follow the movement of another widget.

Remarks:

Inherits from **Ewl_Box** (p. 586).

Data Structures

- struct **Ewl_Floater**

*Inherits from **Ewl_Box** (p. 586) for layout purposes. Provides a means to layer above other widgets and to follow the movement of another widget.*

Defines

- `#define EWL_FLOATER(floater) ((Ewl_Floater *) floater)`
- `#define EWL_FLOATER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FLOATER_TYPE))`
- `#define EWL_FLOATER_TYPE "floater"`

Typedefs

- `typedef Ewl_Floater Ewl_Floater`

Functions

- `void ewl_floater_cb_follow_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_floater_cb_follow_destroy (Ewl_Widget *w, void *ev_data, void *user_data)`
- `Ewl_Widget * ewl_floater_follow_get (Ewl_Floater *f)`
Get the widget we are following.
- `void ewl_floater_follow_set (Ewl_Floater *f, Ewl_Widget *p)`
Set the widget to follow.
- `int ewl_floater_init (Ewl_Floater *f)`
Initialize a floater to default values.
- `Ewl_Widget * ewl_floater_new (void)`
Allocate a new floater widget.
- `void ewl_floater_position_set (Ewl_Floater *parent, int x, int y)`
Set the start x/y position of the floater.

6.32.2 Define Documentation

6.32.2.1 `#define EWL_FLOATER(floater) ((Ewl_Floater *) floater)`

Typecasts a pointer to an **Ewl_Floater** (p. 671) pointer.

6.32.2.2 `#define EWL_FLOATER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FLOATER_TYPE))`

Returns TRUE if the widget is an **Ewl_Floater** (p. 671), FALSE otherwise

6.32.2.3 `#define EWL_FLOATER_TYPE "floater"`

The type name for the **Ewl_Floater** (p. 671) widget

6.32.3 Typedef Documentation

6.32.3.1 `typedef struct Ewl_Floater Ewl_Floater`

The floater is a container for widgets that need to be drawn on top of multiple widgets (ie. dropdownbox)

6.32.4 Function Documentation

6.32.4.1 `void ewl_floater_cb_follow_configure (Ewl_Widget * w, void * ev_data, void * user_data)`

6.32.4.2 `void ewl_floater_cb_follow_destroy (Ewl_Widget * w, void * ev_data, void * user_data)`

6.32.4.3 `Ewl_Widget* ewl_floater_follow_get (Ewl_Floater * f)`

Get the widget we are following.

Parameters:

f,: The Floater to get the follow from

Returns:

Returns the widget the floater is following, or NULL

6.32.4.4 `void ewl_floater_follow_set (Ewl_Floater * f, Ewl_Widget * p)`

Set the widget to follow.

Parameters:

f,: The **Ewl_Floater** (p. 671) to set the follow into

p,: The widget to follow

Returns:

Returns no value

6.32.4.5 int ewl_floater_init (Ewl_Floater * *f*)

Initialize a floater to default values.

Parameters:

f,: the floater widget

Returns:

Returns no value.

Sets the fields and callbacks of the floater *f* to their defaults.

6.32.4.6 Ewl_Widget* ewl_floater_new (void)

Allocate a new floater widget.

Returns:

Returns NULL on failure, or the new floater widget on success.

6.32.4.7 void ewl_floater_position_set (Ewl_Floater * *f*, int *x*, int *y*)

Set the start x/y position of the floater.

Parameters:

f,: **Ewl_Floater** (p. 671) * - the floater widget

x,: int - start x coordinate from the top

y,: int - start y coordinate from the top

Returns:

Returns no value

6.33 Ewl_Freebox: A Freelayout widget

6.33.1 Detailed Description

Defines a class to allow the user to specify the layout type for the widget.

Remarks:

Inherits from **Ewl_Container** (p. 618).

Data Structures

- struct **Ewl_Freebox**

*Inherits from **Ewl_Container** (p. 618) and extends to allow for different layout algorithms.*

Defines

- #define **EWL_FREEBOX**(box) ((**Ewl_Freebox** *)box)
- #define **EWL_FREEBOX_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_FREEBOX_TYPE**))
- #define **EWL_FREEBOX_TYPE** "freebox"

Typedefs

- typedef **Ewl_Freebox** **Ewl_Freebox**
- typedef int(*) **Ewl_Freebox_Comparator** (**Ewl_Widget** *a, **Ewl_Widget** *b)

Functions

- void **ewl_freebox_cb_child_add** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_freebox_cb_child_show** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_freebox_cb_configure** (**Ewl_Widget** *w, void *ev, void *data)
- **Ewl_Freebox_Comparator** **ewl_freebox_comparator_get** (**Ewl_Freebox** *fb)
*Retrieve the comparator set in the **Ewl_Freebox** (p. 673).*
- void **ewl_freebox_comparator_set** (**Ewl_Freebox** *fb, **Ewl_Freebox_Comparator** cmp)
*Set the comparator to use in the **Ewl_Freebox** (p. 673).*
- int **ewl_freebox_init** (**Ewl_Freebox** *fb)
*Initializes an **Ewl_Freebox** (p. 673) widget to default values.*
- **Ewl_Freebox_Layout_Type** **ewl_freebox_layout_type_get** (**Ewl_Freebox** *fb)
*Retrieve the layout type of the **Ewl_Freebox** (p. 673).*
- void **ewl_freebox_layout_type_set** (**Ewl_Freebox** *fb, **Ewl_Freebox_Layout_Type** type)

*Set the layout type of the **Ewl_Freebox** (p. 673).*

- **Ewl_Widget * ewl_freebox_new** (void)
*Creates a new, initialized **Ewl_Freebox** (p. 673) widget.*
- **Ewl_Orientation ewl_freebox_orientation_get** (**Ewl_Freebox** *fb)
Retrieve the current orientation of the freebox.
- **void ewl_freebox_orientation_set** (**Ewl_Freebox** *fb, **Ewl_Orientation** orientation)
Sets the orientation of the freebox.
- **void ewl_freebox_resort** (**Ewl_Freebox** *fb)
This will force the freebox to resort and redraw it's contents. Useful if part of the content changes but no child widgets where added/removed. (An icon label changes for example).
- **Ewl_Widget * ewl_hfreebox_new** (void)
creates and initializes a new freebox widget
- **Ewl_Widget * ewl_vfreebox_new** (void)
creates and initializes a new freebox widget

6.33.2 Define Documentation

6.33.2.1 **#define EWL_FREEBOX(box) ((Ewl_Freebox *)box)**

Typecasts a pointer to an **Ewl_Freebox** (p. 673) pointer.

6.33.2.2 **#define EWL_FREEBOX_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FREEBOX_TYPE))**

Returns TRUE if the widget is an **Ewl_Freebox** (p. 673), FALSE otherwise

6.33.2.3 **#define EWL_FREEBOX_TYPE "freebox"**

The type name for the **Ewl_Freebox** (p. 673) widget

6.33.3 Typedef Documentation

6.33.3.1 **typedef struct Ewl_Freebox Ewl_Freebox**

A free layout container.

6.33.3.2 **typedef int(*) Ewl_Freebox_Comparator(Ewl_Widget *a, Ewl_Widget *b)**

Callback to compare two widgets to determine which is first. If the return is < 0 then a is first. If the return is > 0 then b is first. If the return $== 0$ then they are equal

6.33.4 Function Documentation

6.33.4.1 void ewl_freebox_cb_child_add (Ewl_Container * *c*, Ewl_Widget * *w*)

6.33.4.2 void ewl_freebox_cb_child_show (Ewl_Container * *c*, Ewl_Widget * *w*)

6.33.4.3 void ewl_freebox_cb_configure (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.33.4.4 Ewl_Freebox_Comparator ewl_freebox_comparator_get (Ewl_Freebox * *fb*)

Retrieve the comparator set in the **Ewl_Freebox** (p.673).

Parameters:

fb,: The **Ewl_Freebox** (p.673) to get the comparator from

Returns:

Returns the Ewl_Freebox_Comparator set on the given **Ewl_Freebox** (p.673) or NULL if none set.

6.33.4.5 void ewl_freebox_comparator_set (Ewl_Freebox * *fb*,
Ewl_Freebox_Comparator *cmp*)

Set the comparator to use in the **Ewl_Freebox** (p.673).

Parameters:

fb,: The **Ewl_Freebox** (p.673) to set the comparator on

cmp,: The Ewl_Freebox_Comparator function to set

Returns:

Returns no value

6.33.4.6 int ewl_freebox_init (Ewl_Freebox * *fb*)

Initializes an **Ewl_Freebox** (p.673) widget to default values.

Parameters:

fb,: The **Ewl_Freebox** (p.673) to initialize

Returns:

Returns TRUE on success or FALSE on failure

6.33.4.7 Ewl_Freebox_Layout_Type ewl_freebox_layout_type_get (Ewl_Freebox * *fb*)

Retrieve the layout type of the **Ewl_Freebox** (p. 673).

Parameters:

fb,: The **Ewl_Freebox** (p. 673) to get the layout from

Returns:

Returns the **Ewl_Freebox_Layout_Type** currently set on this freebox

6.33.4.8 void ewl_freebox_layout_type_set (Ewl_Freebox * *fb*, Ewl_Freebox_Layout_Type *type*)

Set the layout type of the **Ewl_Freebox** (p. 673).

Parameters:

fb,: The **Ewl_Freebox** (p. 673) to change

type,: The **Ewl_Freebox_Layout_Type** to set on this freebox

Returns:

Returns no value

6.33.4.9 Ewl_Widget* ewl_freebox_new (void)

Creates a new, initialized **Ewl_Freebox** (p. 673) widget.

Returns:

Returns a new **Ewl_Freebox** (p. 673) on success or NULL on failure

6.33.4.10 Ewl_Orientation ewl_freebox_orientation_get (Ewl_Freebox * *fb*)

Retrieve the current orientation of the freebox.

Parameters:

fb,: The freebox to use

Returns:

Returns the orientation of the freebox

6.33.4.11 void ewl_freebox_orientation_set (Ewl_Freebox * *fb*, Ewl_Orientation *orientation*)

Sets the orientation of the freebox.

Parameters:

fb,: The freebox to use

orientation,: The orientation to set

Returns:

Returns no value

6.33.4.12 void ewl_freebox_resort (Ewl_Freebox * *fb*)

This will force the freebox to resort and redraw it's contents. Useful if part of the content changes but no child widgets where added/removed. (An icon label changes for example).

Parameters:

fb,: The **Ewl_Freebox** (p. 673) to for a resort on

Returns:

Returns no value.

6.33.4.13 Ewl_Widget* ewl_hfreebox_new (void)

creates and initializes a new freebox widget

Returns:

Returns a new horizontal **Ewl_Freebox** (p. 673) widget or NULL on failure

6.33.4.14 Ewl_Widget* ewl_vfreebox_new (void)

creates and initializes a new freebox widget

Returns:

Returns a new vertical **Ewl_Freebox** (p. 673) widget or NULL on failure

6.34 Ewl_Grid The ewl grid widget

6.34.1 Detailed Description

The Ewl Grid widget.

Remarks:

Inherits from **Ewl_Container** (p. 618).

Data Structures

- struct **Ewl_Grid**
*Inherit from **Ewl_Container** (p. 618) and extend to provide a grid layout widget.*
- struct **Ewl_Grid_Child**
Contains information about a grid child.
- struct **Ewl_Grid_Info**
*Contains information about a row or column of **Ewl_Grid** (p. 675).*

Defines

- #define **EWL_GRID**(grid) ((**Ewl_Grid** *)grid)
- #define **EWL_GRID_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_GRID_TYPE))
- #define **EWL_GRID_TYPE** "grid"

Typedefs

- typedef **Ewl_Grid** **Ewl_Grid**
- typedef **Ewl_Grid_Child** **Ewl_Grid_Child**
- typedef **Ewl_Grid_Info** **Ewl_Grid_Info**

Functions

- void **ewl_grid_cb_child_add** (**Ewl_Container** *p, **Ewl_Widget** *c)
- void **ewl_grid_cb_child_remove** (**Ewl_Container** *p, **Ewl_Widget** *c, int idx)
- void **ewl_grid_cb_child_resize** (**Ewl_Container** *p, **Ewl_Widget** *child, int size, **Ewl_Orientation** o)
- void **ewl_grid_cb_child_show** (**Ewl_Container** *p, **Ewl_Widget** *child)
- void **ewl_grid_cb_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_grid_cb_destroy** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_grid_child_position_get** (**Ewl_Grid** *g, **Ewl_Widget** *child, int *start_col, int *end_col, int *start_row, int *end_row)
get the position of a child widget
- void **ewl_grid_child_position_set** (**Ewl_Grid** *g, **Ewl_Widget** *child, int start_col, int end_col, int start_row, int end_row)

Give a child widget a fixed-position in the grid.

- `int ewl_grid_column_current_w_get (Ewl_Grid *g, int col)`
Get the current width of a column.
- `int ewl_grid_column_fixed_w_get (Ewl_Grid *g, int col)`
Get the user set width of a column.
- `void ewl_grid_column_fixed_w_set (Ewl_Grid *g, int col, int width)`
Set the fixed size of a column.
- `void ewl_grid_column_preferred_w_use (Ewl_Grid *g, int col)`
use the preferred size of the column
- `float ewl_grid_column_relative_w_get (Ewl_Grid *g, int col)`
Get the user set relative width of a column.
- `void ewl_grid_column_relative_w_set (Ewl_Grid *g, int col, float relw)`
Set the relative width of a column.
- `void ewl_grid_column_w_remove (Ewl_Grid *g, int col)`
remove the user set size
- `void ewl_grid_dimensions_get (Ewl_Grid *g, int *col, int *row)`
get the number of columns and rows
- `void ewl_grid_dimensions_set (Ewl_Grid *g, int col, int row)`
sets the new dimensions
- `unsigned int ewl_grid_hhomogeneous_get (Ewl_Grid *g)`
Retrieves the horizontal homogeneous flag.
- `void ewl_grid_hhomogeneous_set (Ewl_Grid *g, unsigned int h)`
Change the horizontal homogeneous layout of the box.
- `void ewl_grid_homogeneous_set (Ewl_Grid *g, unsigned int h)`
Change the homogeneous layout of the grid.
- `int ewl_grid_init (Ewl_Grid *g)`
*Initializes an **Ewl_Grid** (p. 675) widget to default values.*
- `Ewl_Widget * ewl_grid_new (void)`
*Create a new **Ewl_Grid** (p. 675) widget.*
- `Ewl_Orientation ewl_grid_orientation_get (Ewl_Grid *g)`
Retrieves the fill orientation flag.
- `void ewl_grid_orientation_set (Ewl_Grid *g, Ewl_Orientation orientation)`
Change the fill orientation.

- `int ewl_grid_row_current_h_get (Ewl_Grid *g, int row)`
Get the current height of a column.
- `int ewl_grid_row_fixed_h_get (Ewl_Grid *g, int row)`
Get the user set height of a row.
- `void ewl_grid_row_fixed_h_set (Ewl_Grid *g, int row, int height)`
Set the fixed size of a column.
- `void ewl_grid_row_h_remove (Ewl_Grid *g, int row)`
remove the user set size
- `void ewl_grid_row_preferred_h_use (Ewl_Grid *g, int col)`
use the preferred size of the row
- `float ewl_grid_row_relative_h_get (Ewl_Grid *g, int col)`
Get the user set relative height of a row.
- `void ewl_grid_row_relative_h_set (Ewl_Grid *g, int col, float relh)`
Set the relative height of a row.
- `unsigned int ewl_grid_vhomogeneous_get (Ewl_Grid *g)`
Retrieves the vertical homogeneous flag.
- `void ewl_grid_vhomogeneous_set (Ewl_Grid *g, unsigned int h)`
Change the vertical homogeneous layout of the box.

6.34.2 Define Documentation

6.34.2.1 `#define EWL_GRID(grid) ((Ewl_Grid *)grid)`

Typecast a pointer to an `Ewl_Grid` (p. 675) pointer

6.34.2.2 `#define EWL_GRID_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_GRID_TYPE))`

Returns TRUE if the widget is an `Ewl_Grid` (p. 675), FALSE otherwise

6.34.2.3 `#define EWL_GRID_TYPE "grid"`

The type name for the `Ewl_Grid` (p. 675) widget

6.34.3 Typedef Documentation

6.34.3.1 `typedef struct Ewl_Grid Ewl_Grid`

The `Ewl_Grid` (p. 675) widget

6.34.3.2 typedef struct Ewl_Grid_Child Ewl_Grid_Child

The `Ewl_Grid_Child` (p. 677) type

6.34.3.3 typedef struct Ewl_Grid_Info Ewl_Grid_Info

The `Ewl_Grid_Info` (p. 678) structure

6.34.4 Function Documentation**6.34.4.1** void ewl_grid_cb_child_add (Ewl_Container * *p*, Ewl_Widget * *c*)**6.34.4.2** void ewl_grid_cb_child_remove (Ewl_Container * *p*, Ewl_Widget * *c*, int *idx*)**6.34.4.3** void ewl_grid_cb_child_resize (Ewl_Container * *p*, Ewl_Widget * *child*, int *size*, Ewl_Orientation *o*)**6.34.4.4** void ewl_grid_cb_child_show (Ewl_Container * *p*, Ewl_Widget * *child*)**6.34.4.5** void ewl_grid_cb_configure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)**6.34.4.6** void ewl_grid_cb_destroy (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)**6.34.4.7** void ewl_grid_child_position_get (Ewl_Grid * *g*, Ewl_Widget * *w*, int * *start_col*, int * *end_col*, int * *start_row*, int * *end_row*)

get the position of a child widget

Parameters:

g,: the grid

w,: the child widget

start_col,: the start column

end_col,: the end column

start_row,: the start row

end_row,: the end row

Returns:

Returns no value

6.34.4.8 void ewl_grid_child_position_set (Ewl_Grid * *g*, Ewl_Widget * *w*, int *start_col*, int *end_col*, int *start_row*, int *end_row*)

Give a child widget a fixed-postion in the grid.

Parameters:

g,: the grid
w,: the child widget
start_col,: the start column
end_col,: the end column
start_row,: the start row
end_row,: the end row

Returns:

Returns no value

6.34.4.9 int ewl_grid_column_current_w_get (Ewl_Grid * *g*, int *col*)

Get the current width of a column.

Parameters:

g,: the grid
col,: the column

Returns:

Returns the current width

This function returns current width of the column.

6.34.4.10 int ewl_grid_column_fixed_w_get (Ewl_Grid * *g*, int *col*)

Get the user set width of a column.

Parameters:

g,: the grid
col,: the column

Returns:

Returns the user set width

This function returns only the size set by the user.

6.34.4.11 void ewl_grid_column_fixed_w_set (Ewl_Grid * *g*, int *col*, int *width*)

Set the fixed size of a column.

Parameters:

g,: the grid
col,: the column
width,: the new width

Returns:

Returns no value.

6.34.4.12 void ewl_grid_column_preferred_w_use (Ewl_Grid * *g*, int *col*)

use the preferred size of the column

Parameters:

g,: the grid
col,: the column

Returns:

Returns no value.

6.34.4.13 float ewl_grid_column_relative_w_get (Ewl_Grid * *g*, int *col*)

Get the user set relative width of a column.

Parameters:

g,: the grid
col,: the column

Returns:

Returns the user set relative width

This function returns only the relative size set by the user.

6.34.4.14 void ewl_grid_column_relative_w_set (Ewl_Grid * *g*, int *col*, float *relw*)

Set the relative width of a column.

Parameters:

g,: the grid
col,: the column
relw,: the new relative width

Returns:

Returns no value.

6.34.4.15 void ewl_grid_column_w_remove (Ewl_Grid * *g*, int *col*)

remove the user set size

Parameters:

g,: the grid
col,: the column

Returns:

Returns no value.

6.34.4.16 void ewl_grid_dimensions_get (Ewl_Grid * *g*, int * *cols*, int * *rows*)

get the number of columns and rows

Parameters:

- g*,: the grid
- cols*,: a pointer where the number of columns will be saved
- rows*,: a pointer where the number of rows will be saved

Returns:

Returns no value

6.34.4.17 void ewl_grid_dimensions_set (Ewl_Grid * *g*, int *cols*, int *rows*)

sets the new dimensions

Parameters:

- g*,: the grid
- cols*,: the new number of columns
- rows*,: the new number of rows

Returns:

Returns no value

6.34.4.18 unsigned int ewl_grid_hhomogeneous_get (Ewl_Grid * *g*)

Retrieves the horizontal homogeneous flag.

Parameters:

- g*,: the grid to get the homogeneous layout

Returns:

The horizontal homogeneous flag

6.34.4.19 void ewl_grid_hhomogeneous_set (Ewl_Grid * *g*, unsigned int *h*)

Change the horizontal homogeneous layout of the box.

Parameters:

- g*,: the grid to change horizontal homogeneous layout
- h*,: the boolean value to change the horizontal layout mode to

Returns:

Returns no value.

Grids use non-homogeneous layout by default, this can be used to change that for horizontal orientation, i.e. all columns can have the same width.

6.34.4.20 void ewl_grid_homogeneous_set (Ewl_Grid * *g*, unsigned int *h*)

Change the homogeneous layout of the grid.

Parameters:

- g*,: the grid to change homogeneous layout
- h*,: the boolean value to change the layout mode to

Returns:

Returns no value.

Grids use non-homogeneous layout by default, this can be used to change that.

6.34.4.21 int ewl_grid_init (Ewl_Grid * *g*)

Initializes an **Ewl_Grid** (p. 675) widget to default values.

Parameters:

- g*,: the grid

Returns:

Returns no value. Responsible for setting up default values and callbacks within a grid structure

6.34.4.22 Ewl_Widget* ewl_grid_new (void)

Create a new **Ewl_Grid** (p. 675) widget.

Returns:

Returns a pointer to a grid on success or NULL on failure.

6.34.4.23 Ewl_Orientation ewl_grid_orientation_get (Ewl_Grid * *g*)

Retrieves the fill orientation flag.

Parameters:

- g*,: the grid to get the fill orientation

Returns:

The fill orientation flag

6.34.4.24 void ewl_grid_orientation_set (Ewl_Grid * *g*, Ewl_Orientation *orientation*)

Change the fill orientation.

Parameters:

g,: the grid to change the fill orientation
orientation,: the Ewl_Orientation value

Returns:

Returns no value.

6.34.4.25 int ewl_grid_row_current_h_get (Ewl_Grid * *g*, int *row*)

Get the current height of a column.

Parameters:

g,: the grid
row,: the row

Returns:

Returns the current height

This function returns current width of the column.

6.34.4.26 int ewl_grid_row_fixed_h_get (Ewl_Grid * *g*, int *row*)

Get the user set height of a row.

Parameters:

g,: the grid
row,: the row

Returns:

Returns the user set height

This function returns only the size set by the user.

6.34.4.27 void ewl_grid_row_fixed_h_set (Ewl_Grid * *g*, int *row*, int *height*)

Set the fixed size of a column.

Parameters:

g,: the grid
row,: the row
height,: the new height

Returns:

Returns no value.

6.34.4.28 void ewl_grid_row_h_remove (Ewl_Grid * *g*, int *row*)

remove the user set size

Parameters:

g,: the grid
row,: the row

Returns:

Returns no value.

6.34.4.29 void ewl_grid_row_preferred_h_use (Ewl_Grid * *g*, int *row*)

use the preferred size of the row

Parameters:

g,: the grid
row,: the row

Returns:

Returns no value.

6.34.4.30 float ewl_grid_row_relative_h_get (Ewl_Grid * *g*, int *row*)

Get the user set relative height of a row.

Parameters:

g,: the grid
row,: the row

Returns:

Returns the user set relative height

This function returns only the relative size set by the user.

6.34.4.31 void ewl_grid_row_relative_h_set (Ewl_Grid * *g*, int *row*, float *relh*)

Set the relative height of a row.

Parameters:

g,: the grid
row,: the row
relh,: the new relative height

Returns:

Returns no value.

6.34.4.32 unsigned int ewl_grid_vhomogeneous_get (Ewl_Grid * *g*)

Retrieves the vertical homogeneous flag.

Parameters:

g,: the grid to get the vertical layout

Returns:

The vertical homogeneous flag

6.34.4.33 void ewl_grid_vhomogeneous_set (Ewl_Grid * *g*, unsigned int *h*)

Change the vertical homogeneous layout of the box.

Parameters:

g,: the grid to change vertical homogeneous layout

h,: the boolean value to change the vertical layout mode to

Returns:

Returns no value.

Grids use non-homogeneous layout by default, this can be used to change that for vertical orientation, i.e. all rows can have the same height.

6.35 Ewl_Highlight: A highlight widget

6.35.1 Detailed Description

Defines a widget to highlight other widgets

Remarks:

Inherits from **Ewl_Floater** (p. 671).

Data Structures

- struct **Ewl_Highlight**

*Inherits from **Ewl_Floater** (p. 671) and extends to provide a highlight widget.*

Defines

- #define **EWL_HIGHLIGHT(w)** ((**Ewl_Highlight ***)w)
- #define **EWL_HIGHLIGHT_IS(w)** (ewl_widget_type_is(**EWL_WIDGET(w)**, **EWL_HIGHLIGHT_TYPE**))
- #define **EWL_HIGHLIGHT_TYPE** "highlight"

Typedefs

- typedef **Ewl_Highlight** **Ewl_Highlight**

Functions

- **Ewl_Widget *** ewl_highlight_follow_get (**Ewl_Highlight ***h)
Retrieves the followed widget for the h highlight.
- void ewl_highlight_follow_set (**Ewl_Highlight ***h, **Ewl_Widget ***w)
Sets the highlight widget to follow the given w widget.
- int ewl_highlight_init (**Ewl_Highlight ***h)
*Initializes an **Ewl_Highlight** (p. 680) widget to default values.*
- **Ewl_Widget *** ewl_highlight_new (void)
Creates and returns a new widget on success or NULL on failure.

6.35.2 Define Documentation

6.35.2.1 #define EWL_HIGHLIGHT(w) ((Ewl_Highlight *)w)

Typecasts a pointer to an **Ewl_Highlight** (p. 680) pointer.

6.35.2.2 `#define EWL_HIGHLIGHT_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_HIGHLIGHT_TYPE))`

Returns TRUE if the widget is an **Ewl_Highlight** (p. 680), FALSE otherwise

6.35.2.3 `#define EWL_HIGHLIGHT_TYPE "highlight"`

The type name for the **Ewl_Highlight** (p. 680) widget

6.35.3 Typedef Documentation

6.35.3.1 `typedef struct Ewl_Highlight Ewl_Highlight`

Ewl_Highlight (p. 680)

6.35.4 Function Documentation

6.35.4.1 `Ewl_Widget* ewl_highlight_follow_get (Ewl_Highlight * h)`

Retrieves the followed widget for the *h* highlight.

Parameters:

h,: The highlight widget to work with

Returns:

Returns the followed widget or NULL

6.35.4.2 `void ewl_highlight_follow_set (Ewl_Highlight * h, Ewl_Widget * w)`

Sets the highlight widget to follow the given *w* widget.

Parameters:

h,: The **Ewl_Highlight** (p. 680) widget to set the follower on

w,: The widget to highlight

Returns:

Returns no value

6.35.4.3 `int ewl_highlight_init (Ewl_Highlight * h)`

Initializes an **Ewl_Highlight** (p. 680) widget to default values.

Parameters:

h,: The **Ewl_Highlight** (p. 680) widget to initialize

Returns:

Returns TRUE on successful initialization, FALSE otherwise

6.35.4.4 Ewl_Widget* ewl_highlight_new (void)

Creates and returns a new widget on success or NULL on failure.

Returns:

Returns a new **Ewl_Widget** (p. 785) on success or NULL on failure

6.36 Ewl_Histogram: A Simple Histogram widget

6.36.1 Detailed Description

Remarks:

Inherits from **Ewl_Image** (p. 686).

Data Structures

- struct **Ewl_Histogram**

*Inherit from the **Ewl_Image** (p. 686) and extends to create a histogram widget.*

Defines

- #define **EWL_HISTOGRAM**(histogram) ((**Ewl_Histogram** *)histogram)
- #define **EWL_HISTOGRAM_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_HISTOGRAM_TYPE))
- #define **EWL_HISTOGRAM_TYPE** "histogram"

Typedefs

- typedef **Ewl_Histogram** **Ewl_Histogram**

Functions

- void **ewl_histogram_cb_configure** (**Ewl_Widget** *w, void *event, void *data)
- **Ewl_Histogram_Channel** **ewl_histogram_channel_get** (**Ewl_Histogram** *histogram)
Sets the color channel to graph in the histogram.
- void **ewl_histogram_channel_set** (**Ewl_Histogram** *histogram, **Ewl_Histogram_Channel** channel)
Sets the color channel to graph in the histogram.
- void **ewl_histogram_color_get** (**Ewl_Histogram** *histogram, int *r, int *g, int *b, int *a)
Get the current color values for drawing the histogram.
- void **ewl_histogram_color_set** (**Ewl_Histogram** *histogram, int r, int g, int b, int a)
Changes the drawing color of a histogram.
- **Ewl_Image** * **ewl_histogram_image_get** (**Ewl_Histogram** *histogram)
Get the source image used to generate the histogram.
- void **ewl_histogram_image_set** (**Ewl_Histogram** *histogram, **Ewl_Image** *image)
Change the source image used to generate the histogram.

- `int ewl_histogram_init (Ewl_Histogram *histogram)`

Initialize a histogram widget to starting values.

- `Ewl_Widget * ewl_histogram_new (void)`

*Create a new **Ewl_Histogram** (p. 681) widget.*

6.36.2 Define Documentation

6.36.2.1 `#define EWL_HISTOGRAM(histogram) ((Ewl_Histogram *)histogram)`

Typecast a pointer to an **Ewl_Histogram** (p. 681) pointer.

6.36.2.2 `#define EWL_HISTOGRAM_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_HISTOGRAM_TYPE))`

Returns TRUE if the widget is an **Ewl_Histogram** (p. 681), FALSE otherwise

6.36.2.3 `#define EWL_HISTOGRAM_TYPE "histogram"`

The type name for the **Ewl_Histogram** (p. 681) widget

6.36.3 Typedef Documentation

6.36.3.1 `typedef struct Ewl_Histogram Ewl_Histogram`

The **Ewl_Histogram** (p. 681)

6.36.4 Function Documentation

6.36.4.1 `void ewl_histogram_cb_configure (Ewl_Widget * w, void * event, void * data)`

6.36.4.2 `Ewl_Histogram_Channel ewl_histogram_channel_get (Ewl_Histogram * hist)`

Sets the color channel to graph in the histogram.

Parameters:

hist,: the histogram to change the channel drawn

Returns:

Returns the currently drawn color channel.

6.36.4.3 `void ewl_histogram_channel_set (Ewl_Histogram * hist,
Ewl_Histogram_Channel channel)`

Sets the color channel to graph in the histogram.

Parameters:

hist,: the histogram to change the channel drawn

channel,: the color channel to draw in the histogram

Returns:

Returns no value.

6.36.4.4 `void ewl_histogram_color_get (Ewl_Histogram * hist, int * r, int * g, int
* b, int * a)`

Get the current color values for drawing the histogram.

Parameters:

hist,: the histogram widget to retrieve display color

r,: red value for histogram drawing color

g,: green value for histogram drawing color

b,: blue value for histogram drawing color

a,: apha value for histogram drawing color

Returns:

Returns no value.

6.36.4.5 `void ewl_histogram_color_set (Ewl_Histogram * hist, int r, int g, int b,
int a)`

Changes the drawing color of a histogram.

Parameters:

hist,: the histogram widget to change display color

r,: red value for histogram drawing color

g,: green value for histogram drawing color

b,: blue value for histogram drawing color

a,: apha value for histogram drawing color

Returns:

Returns no value.

6.36.4.6 Ewl_Image* ewl_histogram_image_get (Ewl_Histogram * *hist*)

Get the source image used to generate the histogram.

Parameters:

hist,: the histogram to get source image

Returns:

Returns a pointer to the current source image.

6.36.4.7 void ewl_histogram_image_set (Ewl_Histogram * *hist*, Ewl_Image * *image*)

Change the source image used to generate the histogram.

Parameters:

hist,: the histogram to change source image

image,: the new source image for the histogram

Returns:

Returns no value.

6.36.4.8 int ewl_histogram_init (Ewl_Histogram * *hist*)

Initialize a histogram widget to starting values.

Parameters:

hist,: the histogram widget to initialize

Returns:

Returns TRUE on success, FALSE on failure.

6.36.4.9 Ewl_Widget* ewl_histogram_new (void)

Create a new **Ewl_Histogram** (p. 681) widget.

Returns:

Returns a new **Ewl_Histogram** (p. 681) widget.

6.37 Ewl_Icon: An icon widget

6.37.1 Detailed Description

Describes a widget to layout and manipulate icons.

Remarks:

Inherits from **Ewl_Stock** (p. 750).

Data Structures

- struct **Ewl_Icon**

*Inherits from **Ewl_Stock** (p. 750) and extends to provide for an icon layout.*

Defines

- #define **EWL_ICON**(icon) ((**Ewl_Icon** *)icon)
- #define **EWL_ICON_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_ICON_TYPE))
- #define **EWL_ICON_TYPE** "icon"

Typedefs

- typedef **Ewl_Icon** **Ewl_Icon**

Functions

- const char * **ewl_icon_alt_text_get** (**Ewl_Icon** *icon)
Retrieves the alternate text set on the icon.
- void **ewl_icon_alt_text_set** (**Ewl_Icon** *icon, const char *txt)
Sets the given text as the alternate text for the icon.
- void **ewl_icon_cb_destroy** (**Ewl_Widget** *w, void *ev, void *data)
- unsigned int **ewl_icon_constrain_get** (**Ewl_Icon** *icon)
Retrieve the constrain value set on the icon.
- void **ewl_icon_constrain_set** (**Ewl_Icon** *icon, unsigned int val)
Set the constrain value on the icon.
- unsigned int **ewl_icon_editable_get** (**Ewl_Icon** *icon)
Retrieve if the icon is editable or not.
- void **ewl_icon_editable_set** (**Ewl_Icon** *icon, unsigned int e)
Set if the icon is editable or not.
- **Ewl_Widget** * **ewl_icon_extended_data_get** (**Ewl_Icon** *icon)

Retrieve the extended data from the icon.

- void **ewl_icon_extended_data_set** (**Ewl_Icon** *icon, **Ewl_Widget** *ext)
Set the extended data into the icon.
- const char * **ewl_icon_image_file_get** (**Ewl_Icon** *icon)
Retrieve the image to used in the icon.
- void **ewl_icon_image_set** (**Ewl_Icon** *icon, const char *file, const char *key)
set the image to use in the icon
- int **ewl_icon_init** (**Ewl_Icon** *icon)
*Initializes the given **Ewl_Icon** (p. 683) widget.*
- unsigned int **ewl_icon_label_complex_get** (**Ewl_Icon** *icon)
Retrieve if the icon has a complex label or not.
- void **ewl_icon_label_complex_set** (**Ewl_Icon** *icon, unsigned int complex_label)
Set if the icon use a label or a text widget for the label part.
- unsigned int **ewl_icon_label_compressed_get** (**Ewl_Icon** *icon)
Retrieves the current compressiion setting for the icon.
- void **ewl_icon_label_compressed_set** (**Ewl_Icon** *icon, unsigned int compress)
Sets the compressions setting for the icon to the given value.
- const char * **ewl_icon_label_get** (**Ewl_Icon** *icon)
Retrieve the label from the icon.
- void **ewl_icon_label_set** (**Ewl_Icon** *icon, const char *label)
Set the label of the icon.
- **Ewl_Widget** * **ewl_icon_menu_get** (**Ewl_Icon** *icon)
Retrieve the menu from the icon.
- void **ewl_icon_menu_set** (**Ewl_Icon** *icon, **Ewl_Widget** *menu)
Set the menu for the icon.
- **Ewl_Widget** * **ewl_icon_new** (void)
*Creates and initializes a new **Ewl_Icon** (p. 683) widget.*
- void **ewl_icon_part_hide** (**Ewl_Icon** *icon, **Ewl_Icon_Part** part)
Hide the given part of the icon. Note: You can only hide one part.
- **Ewl_Widget** * **ewl_icon_simple_new** (void)
*Creates and initializes a new **Ewl_Icon** (p. 683) widget.*
- unsigned int **ewl_icon_thumbnailing_get** (**Ewl_Icon** *icon)
Retrieve if the icon use thumbnailing.

- void **ewl_icon_thumbnailing_set** (**Ewl_Icon** *icon, unsigned int thumb)
Set if the icon thumbnailled the image or use it directly.
- **Ewl_Icon_Type** **ewl_icon_type_get** (**Ewl_Icon** *icon)
Retrieve the type of the icon.
- void **ewl_icon_type_set** (**Ewl_Icon** *icon, **Ewl_Icon_Type** type)
Set the type of the icon.

6.37.2 Define Documentation

6.37.2.1 #define **EWL_ICON**(icon) ((**Ewl_Icon** *)icon)

Typecasts a pointer to an **Ewl_Icon** (p. 683) pointer.

6.37.2.2 #define **EWL_ICON_IS**(w) (**ewl_widget_type_is**(**EWL_WIDGET**(w), **EWL_ICON_TYPE**))

Returns TRUE if the widget is an **Ewl_Icon** (p. 683), FALSE otherwise

6.37.2.3 #define **EWL_ICON_TYPE** "icon"

The type name for the **Ewl_Icon** (p. 683) widget

6.37.3 Typedef Documentation

6.37.3.1 typedef struct **Ewl_Icon** **Ewl_Icon**

A widget to display and manipulate an icon

6.37.4 Function Documentation

6.37.4.1 const char* **ewl_icon_alt_text_get** (**Ewl_Icon** * *icon*)

Retrieves the alternate text set on the icon.

Parameters:

icon,: The icon to work with

Returns:

Returns the alternate text set on the icon

6.37.4.2 void **ewl_icon_alt_text_set** (**Ewl_Icon** * *icon*, const char * *txt*)

Sets the given text as the alternate text for the icon.

Parameters:

icon,: The icon to work with
txt,: The text to set as the alternate text

Returns:

Returns no value

6.37.4.3 void ewl_icon_cb_destroy (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.37.4.4 unsigned int ewl_icon_constrain_get (Ewl_Icon * *icon*)

Retrieve the constrain value set on the icon.

Parameters:

icon,: The icon to get the constrain from

Returns:

Returns the current constrain value of the icon

6.37.4.5 void ewl_icon_constrain_set (Ewl_Icon * *icon*, unsigned int *val*)

Set the constrain value on the icon.

Parameters:

icon,: The Ewl_Icon (p. 683) to constrain
val,: The val to constrain too

Returns:

Returns no value.

6.37.4.6 unsigned int ewl_icon_editable_get (Ewl_Icon * *icon*)

Retrieve if the icon is editable or not.

Parameters:

icon,: The icon to check if it's editable

Returns:

Returns TRUE if the icon is editable, FALSE otherwise

6.37.4.7 void ewl_icon_editable_set (Ewl_Icon * *icon*, unsigned int *e*)

Set if the icon is editable or not.

Parameters:

icon,: The icon to set if it is editable or not
e,: The value to set as the editable flag

Returns:

Returns no value.

6.37.4.8 Ewl_Widget* ewl_icon_extended_data_get (Ewl_Icon * *icon*)

Retrieve the extended data from the icon.

Parameters:

icon,: The icon to get the extended data from

Returns:

Returns the extended data on the icon, or NULL if none set

6.37.4.9 void ewl_icon_extended_data_set (Ewl_Icon * *icon*, Ewl_Widget * *ext*)

Set the extended data into the icon.

Parameters:

icon,: The icon to set the extended data into
ext,: The extended data to set in the icon

Returns:

Returns no value

Note:

The widget passed in here becomes internal to the icon, you should not delete it after this. You can pack widgets as needed but the icon will handle the show/hide of the widget after this

6.37.4.10 const char* ewl_icon_image_file_get (Ewl_Icon * *icon*)

Retrieve the image to used in the icon.

Parameters:

icon,: The **Ewl_Icon** (p. 683) to get the image file from

Returns:

Returns the image file associated with this icon, or NULL if none.

6.37.4.11 void ewl_icon_image_set (Ewl_Icon * *icon*, const char * *file*, const char * *key*)

set the image to use in the icon

Parameters:

icon,: The **Ewl_Icon** (p. 683) to set the image into

file,: The file with the image

key,: The key inside the file if applicable

Returns:

Returns no value

6.37.4.12 int ewl_icon_init (Ewl_Icon * *icon*)

Initializes the given **Ewl_Icon** (p. 683) widget.

Parameters:

icon,: The widget to initialize

Returns:

Returns TRUE on successful initialization, FALSE otherwise

6.37.4.13 unsigned int ewl_icon_label_complex_get (Ewl_Icon * *icon*)

Retrieve if the icon has a complex label or not.

Parameters:

icon,: The icon to check if its label is complex

Returns:

Returns TRUE if the label is complex, FALSE otherwise

6.37.4.14 void ewl_icon_label_complex_set (Ewl_Icon * *icon*, unsigned int *c*)

Set if the icon use a label or a text widget for the label part.

Parameters:

icon,: The icon to set if it has a complex label or not

c,: The value to set as the complex flag

Returns:

Returns no value.

6.37.4.15 `unsigned int ewl_icon_label_compressed_get (Ewl_Icon * icon)`

Retrieves the current compression setting for the icon.

Parameters:

icon,: The ewl_icon to work with

Returns:

Returns the current compression setting for the icon

6.37.4.16 `void ewl_icon_label_compressed_set (Ewl_Icon * icon, unsigned int compress)`

Sets the compressions setting for the icon to the given value.

Parameters:

icon,: The icon to work with

compress,: The compression setting to use

Returns:

Returns no value

6.37.4.17 `const char* ewl_icon_label_get (Ewl_Icon * icon)`

Retrieve the label from the icon.

Parameters:

icon,: The icon to get the label from

Returns:

Returns the icons label or NULL if none set

6.37.4.18 `void ewl_icon_label_set (Ewl_Icon * icon, const char * label)`

Set the label of the icon.

Parameters:

icon,: The icon to set the label onto

label,: The label to set on the icon

Returns:

Returns no value

6.37.4.19 Ewl_Widget* ewl_icon_menu_get (Ewl_Icon * *icon*)

Retrieve the menu from the icon.

Parameters:

icon,: The icon to get the menu from

Returns:

Returns the menu set on this icon, or NULL if none set

6.37.4.20 void ewl_icon_menu_set (Ewl_Icon * *icon*, Ewl_Widget * *menu*)

Set the menu for the icon.

Parameters:

icon,: The icon to set the menu into

menu,: The menu to set on the icon

Returns:

Returns no value

6.37.4.21 Ewl_Widget* ewl_icon_new (void)

Creates and initializes a new **Ewl_Icon** (p. 683) widget.

Returns:

Returns a new **Ewl_Icon** (p. 683) widget, or NULL on failure

6.37.4.22 void ewl_icon_part_hide (Ewl_Icon * *icon*, Ewl_Icon_Part *part*)

Hide the given part of the icon. Note: You can only hide one part.

Parameters:

icon,: The icon to set the part to hide of

part,: The part to hide

Returns:

Returns no value.

6.37.4.23 Ewl_Widget* ewl_icon_simple_new (void)

Creates and initializes a new **Ewl_Icon** (p. 683) widget.

Returns:

Returns a new **Ewl_Icon** (p. 683) widget, or NULL on failure

The difference to **ewl_icon_new()** (p. 243) is that it has lighter default values. The defaults are complex_label: no compressed_label: no editable: no thumbnailing: no

6.37.4.24 unsigned int ewl_icon_thumbnailing_get (Ewl_Icon * *icon*)

Retrieve if the icon use thumbnailing.

Parameters:

icon,: The icon to check if it's thumbnails the image

Returns:

Returns TRUE if the label use thumbnailing, FALSE otherwise

6.37.4.25 void ewl_icon_thumbnailing_set (Ewl_Icon * *icon*, unsigned int *t*)

Set if the icon thumbnailed the image or use it directly.

Parameters:

icon,: The icon to set if the image gets thumbnailed

t,: The value to set as the thumbnail flag

Returns:

Returns no value.

6.37.4.26 Ewl_Icon_Type ewl_icon_type_get (Ewl_Icon * *icon*)

Retrieve the type of the icon.

Parameters:

icon,: The icon to get the type from

Returns:

Returns the Ewl_Icon_Type of the icon

6.37.4.27 void ewl_icon_type_set (Ewl_Icon * *icon*, Ewl_Icon_Type *type*)

Set the type of the icon.

Parameters:

icon,: The icon to set the type of

type The type to set on this icon

Returns:

Returns no value.

6.38 Ewl_Icon_Theme: The icon theme code

6.38.1 Detailed Description

Provides code to retrieve the path to icon theme icons.

Defines

- `#define EWL_ICON_ACCESSORIES_CALCULATOR` "accessories-calculator"
- `#define EWL_ICON_ACCESSORIES_CHARACTER_MAP` "accessories-character-map"
- `#define EWL_ICON_ACCESSORIES_DICTIONARY` "accessories-dictionary"
- `#define EWL_ICON_ACCESSORIES_TEXT_EDITOR` "accessories-text-editor"
- `#define EWL_ICON_ADDRESS_BOOK_NEW` "address-book-new"
- `#define EWL_ICON_APPLICATION_EXIT` "application-exit"
- `#define EWL_ICON_APPLICATION_X_EXECUTABLE` "application-x-executable"
- `#define EWL_ICON_APPLICATIONS_ACCESSORIES` "applications-accessories"
- `#define EWL_ICON_APPLICATIONS_DEVELOPMENT` "applications-development"
- `#define EWL_ICON_APPLICATIONS_GAMES` "applications-games"
- `#define EWL_ICON_APPLICATIONS_GRAPHICS` "applications-graphics"
- `#define EWL_ICON_APPLICATIONS_INTERNET` "applications-internet"
- `#define EWL_ICON_APPLICATIONS_MULTIMEDIA` "applications-multimedia"
- `#define EWL_ICON_APPLICATIONS_OFFICE` "applications-office"
- `#define EWL_ICON_APPLICATIONS_OTHER` "applications-other"
- `#define EWL_ICON_APPLICATIONS_SYSTEM` "applications-system"
- `#define EWL_ICON_APPLICATIONS_UTILITIES` "applications-utilities"
- `#define EWL_ICON_APPOINTMENT_MISSED` "appointment-missed"
- `#define EWL_ICON_APPOINTMENT_NEW` "appointment-new"
- `#define EWL_ICON_APPOINTMENT_SOON` "appointment-soon"
- `#define EWL_ICON_AUDIO_CARD` "audio-card"
- `#define EWL_ICON_AUDIO_INPUT_MICROPHONE` "audio-input-microphone"
- `#define EWL_ICON_AUDIO_VOLUME_HIGH` "audio-volume-high"
- `#define EWL_ICON_AUDIO_VOLUME_LOW` "audio-volume-low"
- `#define EWL_ICON_AUDIO_VOLUME_MEDIUM` "audio-volume-medium"
- `#define EWL_ICON_AUDIO_VOLUME_MUTED` "audio-volume-muted"
- `#define EWL_ICON_AUDIO_X_GENERIC` "audio-x-generic"
- `#define EWL_ICON_BATTERY` "battery"
- `#define EWL_ICON_BATTERY_CAUTION` "battery-caution"
- `#define EWL_ICON_BATTERY_LOW` "battery-low"
- `#define EWL_ICON_CAMERA_PHOTO` "camera-photo"
- `#define EWL_ICON_CAMERA_VIDEO` "camera-video"
- `#define EWL_ICON_COMPUTER` "computer"
- `#define EWL_ICON_CONTACT_NEW` "contact-new"
- `#define EWL_ICON_DIALOG_APPLY` "dialog-apply"
- `#define EWL_ICON_DIALOG_CANCEL` "dialog-cancel"

- `#define EWL_ICON_DIALOG_CLOSE "dialog-close"`
- `#define EWL_ICON_DIALOG_ERROR "dialog-error"`
- `#define EWL_ICON_DIALOG_INFORMATION "dialog-information"`
- `#define EWL_ICON_DIALOG_OK "dialog-ok"`
- `#define EWL_ICON_DIALOG_PASSWORD "dialog-password"`
- `#define EWL_ICON_DIALOG_QUESTION "dialog-question"`
- `#define EWL_ICON_DIALOG_WARNING "dialog-warning"`
- `#define EWL_ICON_DOCUMENT_NEW "document-new"`
- `#define EWL_ICON_DOCUMENT_OPEN "document-open"`
- `#define EWL_ICON_DOCUMENT_OPEN_RECENT "document-open-recent"`
- `#define EWL_ICON_DOCUMENT_PAGE_SETUP "document-page-setup"`
- `#define EWL_ICON_DOCUMENT_PRINT "document-print"`
- `#define EWL_ICON_DOCUMENT_PRINT_PREVIEW "document-print-preview"`
- `#define EWL_ICON_DOCUMENT_PROPERTIES "document-properties"`
- `#define EWL_ICON_DOCUMENT_REVERT "document-revert"`
- `#define EWL_ICON_DOCUMENT_SAVE "document-save"`
- `#define EWL_ICON_DOCUMENT_SAVE_AS "document-save-as"`
- `#define EWL_ICON_DRIVE_CDROM "drive-cdrom"`
- `#define EWL_ICON_DRIVE_HARDDISK "drive-harddisk"`
- `#define EWL_ICON_DRIVE_REMOVABLE_MEDIA "drive-removable-media"`
- `#define EWL_ICON_EDIT_COPY "edit-copy"`
- `#define EWL_ICON_EDIT_CUT "edit-cut"`
- `#define EWL_ICON_EDIT_DELETE "edit-delete"`
- `#define EWL_ICON_EDIT_FIND "edit-find"`
- `#define EWL_ICON_EDIT_FIND_REPLACE "edit-find-replace"`
- `#define EWL_ICON_EDIT_PASTE "edit-paste"`
- `#define EWL_ICON_EDIT_REDO "edit-redo"`
- `#define EWL_ICON_EDIT_SELECT_ALL "edit-select-all"`
- `#define EWL_ICON_EDIT_UNDO "edit-undo"`
- `#define EWL_ICON_EMBLEM_DEFAULT "emblem-default"`
- `#define EWL_ICON_EMBLEM_DOCUMENTS "emblem-documents"`
- `#define EWL_ICON_EMBLEM_DOWNLOADS "emblem-downloads"`
- `#define EWL_ICON_EMBLEM_FAVORITE "emblem-favorite"`
- `#define EWL_ICON_EMBLEM_IMPORTANT "emblem-important"`
- `#define EWL_ICON_EMBLEM_MAIL "emblem-mail"`
- `#define EWL_ICON_EMBLEM_PHOTOS "emblem-photos"`
- `#define EWL_ICON_EMBLEM_READONLY "emblem-readonly"`
- `#define EWL_ICON_EMBLEM_SHARED "emblem-shared"`
- `#define EWL_ICON_EMBLEM_SYMBOLIC_LINK "emblem-symbolic-link"`
- `#define EWL_ICON_EMBLEM_SYNCHRONIZED "emblem-synchronized"`
- `#define EWL_ICON_EMBLEM_SYSTEM "emblem-system"`
- `#define EWL_ICON_EMBLEM_UNREADABLE "emblem-unreadable"`
- `#define EWL_ICON_FACE_ANGEL "face-angel"`
- `#define EWL_ICON_FACE_CRYING "face-crying"`
- `#define EWL_ICON_FACE_DEVIL_GRIN "face-devil-grin"`
- `#define EWL_ICON_FACE_DEVIL_SAD "face-devil-sad"`
- `#define EWL_ICON_FACE_GLASSES "face-glasses"`
- `#define EWL_ICON_FACE_KISS "face-kiss"`
- `#define EWL_ICON_FACE_MONKEY "face-monkey"`

```

• #define EWL_ICON_FACE_PLAIN "face-plain"
• #define EWL_ICON_FACE_SAD "face-sad"
• #define EWL_ICON_FACE_SMILE "face-smile"
• #define EWL_ICON_FACE_SMILE_BIG "face-smile-big"
• #define EWL_ICON_FACE_SMIRK "face-smirk"
• #define EWL_ICON_FACE_SURPRISE "face-surprise"
• #define EWL_ICON_FACE_WINK "face-wink"
• #define EWL_ICON_FOLDER "folder"
• #define EWL_ICON_FOLDER_DRAG_ACCEPT "folder-drag-accept"
• #define EWL_ICON_FOLDER_OPEN "folder-open"
• #define EWL_ICON_FOLDER_REMOTE "folder-remote"
• #define EWL_ICON_FOLDER_VISITING "folder-visiting"
• #define EWL_ICON_FONT_X_GENERIC "font-x-generic"
• #define EWL_ICON_FORMAT_INDENT_LESS "format-indent-less"
• #define EWL_ICON_FORMAT_INDENT_MORE "format-indent-more"
• #define EWL_ICON_FORMAT_JUSTIFY_CENTER "format-justify-center"
• #define EWL_ICON_FORMAT_JUSTIFY_FILL "format-justify-fill"
• #define EWL_ICON_FORMAT_JUSTIFY_LEFT "format-justify-left"
• #define EWL_ICON_FORMAT_JUSTIFY_RIGHT "format-justify-right"
• #define EWL_ICON_FORMAT_TEXT_BOLD "format-text-bold"
• #define EWL_ICON_FORMAT_TEXT_DIRECTION_LTR "format-text-direction-ltr"
• #define EWL_ICON_FORMAT_TEXT_DIRECTION_RTL "format-text-direction-rtl"
• #define EWL_ICON_FORMAT_TEXT_ITALIC "format-text-italic"
• #define EWL_ICON_FORMAT_TEXT_STRIKETHROUGH "format-text-strikethrough"
• #define EWL_ICON_FORMAT_TEXT_UNDERLINE "format-text-underline"
• #define EWL_ICON_GO_BOTTOM "go-bottom"
• #define EWL_ICON_GO_DOWN "go-down"
• #define EWL_ICON_GO_FIRST "go-first"
• #define EWL_ICON_GO_HOME "go-home"
• #define EWL_ICON_GO_JUMP "go-jump"
• #define EWL_ICON_GO_LAST "go-last"
• #define EWL_ICON_GO_NEXT "go-next"
• #define EWL_ICON_GO_PREVIOUS "go-previous"
• #define EWL_ICON_GO_TOP "go-top"
• #define EWL_ICON_GO_UP "go-up"
• #define EWL_ICON_HELP_ABOUT "help-about"
• #define EWL_ICON_HELP_BROWSER "help-browser"
• #define EWL_ICON_HELP_CONTENTS "help-contents"
• #define EWL_ICON_HELP_FAQ "help-faq"
• #define EWL_ICON_IMAGE_LOADING "image-loading"
• #define EWL_ICON_IMAGE_MISSING "image-missing"
• #define EWL_ICON_IMAGE_X_GENERIC "image-x-generic"
• #define EWL_ICON_INPUT_GAMING "input-gaming"
• #define EWL_ICON_INPUT_KEYBOARD "input-keyboard"
• #define EWL_ICON_INPUT_MOUSE "input-mouse"
• #define EWL_ICON_INSERT_IMAGE "insert-image"
• #define EWL_ICON_INSERT_LINK "insert-link"

```

```

• #define EWL_ICON_INSERT_OBJECT "insert-object"
• #define EWL_ICON_INSERT_TEXT "insert-text"
• #define EWL_ICON_LIST_ADD "list-add"
• #define EWL_ICON_LIST_REMOVE "list-remove"
• #define EWL_ICON_MAIL_ATTACHMENT "mail-attachment"
• #define EWL_ICON_MAIL_FORWARD "mail-forward"
• #define EWL_ICON_MAIL_MARK_IMPORTANT "mail-mark-important"
• #define EWL_ICON_MAIL_MARK_JUNK "mail-mark-junk"
• #define EWL_ICON_MAIL_MARK_NOTJUNK "mail-mark-notjunk"
• #define EWL_ICON_MAIL_MARK_READ "mail-mark-read"
• #define EWL_ICON_MAIL_MARK_UNREAD "mail-mark-unread"
• #define EWL_ICON_MAIL_MESSAGE_NEW "mail-message-new"
• #define EWL_ICON_MAIL_READ "mail-read"
• #define EWL_ICON_MAIL_REPLIED "mail-replied"
• #define EWL_ICON_MAIL_REPLY_ALL "mail-reply-all"
• #define EWL_ICON_MAIL_REPLY_SENDER "mail-reply-sender"
• #define EWL_ICON_MAIL_SEND_RECEIVE "mail-send-receive"
• #define EWL_ICON_MAIL_SIGNED "mail-signed"
• #define EWL_ICON_MAIL_SIGNED_VERIFIED "mail-signed-verified"
• #define EWL_ICON_MAIL_UNREAD "mail-unread"
• #define EWL_ICON_MEDIA_CDROM "media-cdrom"
• #define EWL_ICON_MEDIA_EJECT "media-eject"
• #define EWL_ICON_MEDIA_FLOPPY "media-floppy"
• #define EWL_ICON_MEDIA_PLAYBACK_PAUSE "media-playback-pause"
• #define EWL_ICON_MEDIA_PLAYBACK_START "media-playback-start"
• #define EWL_ICON_MEDIA_PLAYBACK_STOP "media-playback-stop"
• #define EWL_ICON_MEDIA_PLAYLIST_REPEAT "media-playlist-repeat"
• #define EWL_ICON_MEDIA_PLAYLIST_SHUFFLE "media-playlist-shuffle"
• #define EWL_ICON_MEDIA_RECORD "media-record"
• #define EWL_ICON_MEDIA_SEEK_BACKWARD "media-seek-backward"
• #define EWL_ICON_MEDIA_SEEK_FORWARD "media-seek-forward"
• #define EWL_ICON_MEDIA_SKIP_BACKWARD "media-skip-backward"
• #define EWL_ICON_MEDIA_SKIP_FORWARD "media-skip-forward"
• #define EWL_ICON_MULTIMEDIA_PLAYER "multimedia-player"
• #define EWL_ICON_MULTIMEDIA_VOLUME_CONTROL "multimedia-
volume-control"
• #define EWL_ICON_NETWORK_ERROR "network-error"
• #define EWL_ICON_NETWORK_IDLE "network-idle"
• #define EWL_ICON_NETWORK_OFFLINE "network-offline"
• #define EWL_ICON_NETWORK_RECEIVE "network-receive"
• #define EWL_ICON_NETWORK_SERVER "network-server"
• #define EWL_ICON_NETWORK_TRANSMIT "network-transmit"
• #define EWL_ICON_NETWORK_TRANSMIT_RECEIVE "network-transmit-
receive"
• #define EWL_ICON_NETWORK_WIRED "network-wired"
• #define EWL_ICON_NETWORK_WIRELESS "network-wireless"
• #define EWL_ICON_NETWORK_WORKGROUP "network-workgroup"
• #define EWL_ICON_PACKAGE_X_GENERIC "package-x-generic"
• #define EWL_ICON_PREFERENCES_DESKTOP "preferences-desktop"

```

```

• #define EWL_ICON_PREFERENCES_DESKTOP_ -
ACCESSIBILITY "preferences-desktop-accessibility"
• #define EWL_ICON_PREFERENCES_DESKTOP_ -
ACCESSIBILITY "preferences-desktop-accessibility"
• #define EWL_ICON_PREFERENCES_DESKTOP_FONT "preferences-desktop-
font"
• #define EWL_ICON_PREFERENCES_DESKTOP_KEYBOARD "preferences-
desktop-keyboard"
• #define EWL_ICON_PREFERENCES_DESKTOP_LOCALE "preferences-
desktop-locale"
• #define EWL_ICON_PREFERENCES_DESKTOP_ -
MULTIMEDIA "preferences-desktop-multimedia"
• #define EWL_ICON_PREFERENCES_DESKTOP_ -
PERIPHERALS "preferences-desktop-peripherals"
• #define EWL_ICON_PREFERENCES_DESKTOP_PERSONAL "preferences-
desktop-personal"
• #define EWL_ICON_PREFERENCES_DESKTOP_ -
SCREENSAVER "preferences-desktop-screensaver"
• #define EWL_ICON_PREFERENCES_DESKTOP_THEME "preferences-
desktop-theme"
• #define EWL_ICON_PREFERENCES_DESKTOP_ -
WALLPAPER "preferences-desktop-wallpaper"
• #define EWL_ICON_PREFERENCES_OTHER "preferences-other"
• #define EWL_ICON_PREFERENCES_SYSTEM "preferences-system"
• #define EWL_ICON_PREFERENCES_SYSTEM_NETWORK "preferences-
system-network"
• #define EWL_ICON_PRINTER "printer"
• #define EWL_ICON_PRINTER_ERROR "printer-error"
• #define EWL_ICON_PRINTER_PRINTING "printer-printing"
• #define EWL_ICON_PROCESS_WORKING "process-working"
• #define EWL_ICON_SIZE_LARGE 24
• #define EWL_ICON_SIZE_MEDIUM 22
• #define EWL_ICON_SIZE_SMALL 16
• #define EWL_ICON_SOFTWARE_UPDATE_AVAILABLE "software-update-
available"
• #define EWL_ICON_SOFTWARE_UPDATE_URGENT "software-update-
urgent"
• #define EWL_ICON_START_HERE "start-here"
• #define EWL_ICON_SYNC_ERROR "sync-error"
• #define EWL_ICON_SYNC_SYNCHRONIZING "sync-synchronizing"
• #define EWL_ICON_SYSTEM_FILE_MANAGER "system-file-manager"
• #define EWL_ICON_SYSTEM_HELP "system-help"
• #define EWL_ICON_SYSTEM_LOCK_SCREEN "system-lock-screen"
• #define EWL_ICON_SYSTEM_LOG_OUT "system-log-out"
• #define EWL_ICON_SYSTEM_RUN "system-run"
• #define EWL_ICON_SYSTEM_SEARCH "system-search"
• #define EWL_ICON_SYSTEM_SOFTWARE_UPDATE "system-software-
update"
• #define EWL_ICON_TASK_DUE "task-due"
• #define EWL_ICON_TASK_PASSED_DUE "task-passed-due"
• #define EWL_ICON_TEXT_HTML "text-html"

```

- `#define EWL_ICON_TEXT_X_GENERIC "text-x-generic"`
- `#define EWL_ICON_TEXT_X_GENERIC_TEMPLATE "text-x-generic-template"`
- `#define EWL_ICON_TEXT_X_SCRIPT "text-x-script"`
- `#define EWL_ICON_TOOLS_CHECK_SPELLING "tools-check-spelling"`
- `#define EWL_ICON_USER_AWAY "user-away"`
- `#define EWL_ICON_USER_DESKTOP "user-desktop"`
- `#define EWL_ICON_USER_HOME "user-home"`
- `#define EWL_ICON_USER_IDLE "user-idle"`
- `#define EWL_ICON_USER_OFFLINE "user-offline"`
- `#define EWL_ICON_USER_ONLINE "user-online"`
- `#define EWL_ICON_USER_TRASH "user-trash"`
- `#define EWL_ICON_USER_TRASH_FULL "user-trash-full"`
- `#define EWL_ICON_UTILITIES_TERMINAL "utilities-terminal"`
- `#define EWL_ICON_VIDEO_X_GENERIC "video-x-generic"`
- `#define EWL_ICON_VIEW_FULLSCREEN "view-fullscreen"`
- `#define EWL_ICON_VIEW_REFRESH "view-refresh"`
- `#define EWL_ICON_VIEW_SORT_ASCENDING "view-sort-ascending"`
- `#define EWL_ICON_VIEW_SORT_DESCENDING "view-sort-descending"`
- `#define EWL_ICON_WEATHER_CLEAR "weather-clear"`
- `#define EWL_ICON_WEATHER_CLEAR_NIGHT "weather-clear-night"`
- `#define EWL_ICON_WEATHER_FEW_CLOUDS "weather-few-clouds"`
- `#define EWL_ICON_WEATHER_FEW_CLOUDS_NIGHT "weather-few-clouds-night"`
- `#define EWL_ICON_WEATHER_FOG "weather-fog"`
- `#define EWL_ICON_WEATHER_OVERCAST "weather-overcast"`
- `#define EWL_ICON_WEATHER_SEVERE_ALERT "weather-severe-alert"`
- `#define EWL_ICON_WEATHER_SHOWERS "weather-showers"`
- `#define EWL_ICON_WEATHER_SHOWERS_SCATTERED "weather-showers-scattered"`
- `#define EWL_ICON_WEATHER_SNOW "weather-snow"`
- `#define EWL_ICON_WEATHER_STORM "weather-storm"`
- `#define EWL_ICON_WINDOW_CLOSE "window-close"`
- `#define EWL_ICON_WINDOW_NEW "window-new"`
- `#define EWL_ICON_X_OFFICE_ADDRESS_BOOK "x-office-address-book"`
- `#define EWL_ICON_X_OFFICE_CALENDAR "x-office-calendar"`
- `#define EWL_ICON_X_OFFICE_DOCUMENT "x-office-document"`
- `#define EWL_ICON_X_OFFICE_PRESENTATION "x-office-presentation"`
- `#define EWL_ICON_X_OFFICE_SPREADSHEET "x-office-spreadsheet"`
- `#define EWL_ICON_ZOOM_BEST_FIT "zoom-best-fit"`
- `#define EWL_ICON_ZOOM_IN "zoom-in"`
- `#define EWL_ICON_ZOOM_ORIGINAL "zoom-original"`
- `#define EWL_ICON_ZOOM_OUT "zoom-out"`

Functions

- `const char * ewl_icon_theme_icon_path_get` (`const char *icon`, `int size`)
Retrives the full path to the specified icon, or NULL if none found.
- `int ewl_icon_theme_init` (`void`)
Initializes the icon theme system.
- `void ewl_icon_theme_shutdown` (`void`)
Shuts down the icon theme system.
- `void ewl_icon_theme_theme_change` (`void`)
Called when the icon theme is changed so we can clean up any caching we have in place.

6.38.2 Define Documentation

6.38.2.1 `#define EWL_ICON_ACCESSORIES_CALCULATOR "accessories-calculator"`

accessories calculator

6.38.2.2 `#define EWL_ICON_ACCESSORIES_CHARACTER_MAP "accessories-character-map"`

accessories character map

6.38.2.3 `#define EWL_ICON_ACCESSORIES_DICTIONARY "accessories-dictionary"`

accessories dictionary

6.38.2.4 `#define EWL_ICON_ACCESSORIES_TEXT_EDITOR "accessories-text-editor"`

accessories text editor

6.38.2.5 `#define EWL_ICON_ADDRESS_BOOK_NEW "address-book-new"`

address book icon

6.38.2.6 `#define EWL_ICON_APPLICATION_EXIT "application-exit"`

application exit

6.38.2.7 `#define EWL_ICON_APPLICATION_X_EXECUTABLE "application-x-executable"`

application/x-executable

6.38.2.8 `#define EWL_ICON_APPLICATIONS_ACCESSORIES "applications-accessories"`

apps accessories

6.38.2.9 `#define EWL_ICON_APPLICATIONS_DEVELOPMENT "applications-development"`

apps development

6.38.2.10 `#define EWL_ICON_APPLICATIONS_GAMES "applications-games"`

apps games

6.38.2.11 `#define EWL_ICON_APPLICATIONS_GRAPHICS "applications-graphics"`

apps graphics

6.38.2.12 `#define EWL_ICON_APPLICATIONS_INTERNET "applications-internet"`

apps internet

6.38.2.13 `#define EWL_ICON_APPLICATIONS_MULTIMEDIA "applications-multimedia"`

apps multimedia

6.38.2.14 `#define EWL_ICON_APPLICATIONS_OFFICE "applications-office"`

apps office

6.38.2.15 `#define EWL_ICON_APPLICATIONS_OTHER "applications-other"`

apps other

6.38.2.16 `#define EWL_ICON_APPLICATIONS_SYSTEM "applications-system"`

apps system

6.38.2.17 `#define EWL_ICON_APPLICATIONS_UTILITIES "applications-utilities"`

apps utilities

6.38.2.18 `#define EWL_ICON_APPOINTMENT_MISSED "appointment-missed"`

appointment missed

6.38.2.19 `#define EWL_ICON_APPOINTMENT_NEW "appointment-new"`

appointment new

6.38.2.20 `#define EWL_ICON_APPOINTMENT_SOON "appointment-soon"`

appointment soon

6.38.2.21 `#define EWL_ICON_AUDIO_CARD "audio-card"`

audio card

6.38.2.22 `#define EWL_ICON_AUDIO_INPUT_MICROPHONE "audio-input-microphone"`

audio input microphone

6.38.2.23 `#define EWL_ICON_AUDIO_VOLUME_HIGH "audio-volume-high"`

audio volume high

6.38.2.24 `#define EWL_ICON_AUDIO_VOLUME_LOW "audio-volume-low"`

audio volume low

6.38.2.25 `#define EWL_ICON_AUDIO_VOLUME_MEDIUM "audio-volume-medium"`

audio volume medium

6.38.2.26 `#define EWL_ICON_AUDIO_VOLUME_MUTED "audio-volume-muted"`

audio volume muted

6.38.2.27 `#define EWL_ICON_AUDIO_X_GENERIC "audio-x-generic"`

audio/x-generic

6.38.2.28 `#define EWL_ICON_BATTERY "battery"`

battery

6.38.2.29 `#define EWL_ICON_BATTERY_CAUTION "battery-caution"`

battery caution

6.38.2.30 `#define EWL_ICON_BATTERY_LOW "battery-low"`

battery low

6.38.2.31 `#define EWL_ICON_CAMERA_PHOTO "camera-photo"`

camera photo

6.38.2.32 `#define EWL_ICON_CAMERA_VIDEO "camera-video"`

camera video

6.38.2.33 `#define EWL_ICON_COMPUTER "computer"`

computer

6.38.2.34 `#define EWL_ICON_CONTACT_NEW "contact-new"`

contact new

6.38.2.35 `#define EWL_ICON_DIALOG_APPLY "dialog-apply"`

dialog apply

6.38.2.36 `#define EWL_ICON_DIALOG_CANCEL "dialog-cancel"`

dialog cancel

6.38.2.37 `#define EWL_ICON_DIALOG_CLOSE "dialog-close"`

dialog close

6.38.2.38 `#define EWL_ICON_DIALOG_ERROR "dialog-error"`

dialog error

6.38.2.39 `#define EWL_ICON_DIALOG_INFORMATION "dialog-information"`

dialog information

6.38.2.40 `#define EWL_ICON_DIALOG_OK "dialog-ok"`

dialog ok

6.38.2.41 `#define EWL_ICON_DIALOG_PASSWORD "dialog-password"`

dialog password

6.38.2.42 `#define EWL_ICON_DIALOG_QUESTION "dialog-question"`

dialog question

6.38.2.43 `#define EWL_ICON_DIALOG_WARNING "dialog-warning"`

dialog warning

6.38.2.44 `#define EWL_ICON_DOCUMENT_NEW "document-new"`

document new

6.38.2.45 `#define EWL_ICON_DOCUMENT_OPEN "document-open"`

document open

6.38.2.46 `#define EWL_ICON_DOCUMENT_OPEN_RECENT "document-open-recent"`

document open recent

6.38.2.47 `#define EWL_ICON_DOCUMENT_PAGE_SETUP "document-page-setup"`

document page setup

6.38.2.48 `#define EWL_ICON_DOCUMENT_PRINT "document-print"`

document print

6.38.2.49 `#define EWL_ICON_DOCUMENT_PRINT_PREVIEW "document-print-preview"`

document print preview

6.38.2.50 `#define EWL_ICON_DOCUMENT_PROPERTIES "document-properties"`

document properties

6.38.2.51 `#define EWL_ICON_DOCUMENT_REVERT "document-revert"`

document revert

6.38.2.52 `#define EWL_ICON_DOCUMENT_SAVE "document-save"`

document save

6.38.2.53 `#define EWL_ICON_DOCUMENT_SAVE_AS "document-save-as"`

document save as

6.38.2.54 `#define EWL_ICON_DRIVE_CDROM "drive-cdrom"`

drive cdrom

6.38.2.55 `#define EWL_ICON_DRIVE_HARDDISK "drive-harddisk"`

drive harddisk

6.38.2.56 `#define EWL_ICON_DRIVE_REMOVABLE_MEDIA "drive-removable-media"`

drive removable media

6.38.2.57 `#define EWL_ICON_EDIT_COPY "edit-copy"`

edit copy

6.38.2.58 `#define EWL_ICON_EDIT_CUT "edit-cut"`

edit cut

6.38.2.59 `#define EWL_ICON_EDIT_DELETE "edit-delete"`

edit delete

6.38.2.60 `#define EWL_ICON_EDIT_FIND "edit-find"`

edit find

6.38.2.61 `#define EWL_ICON_EDIT_FIND_REPLACE "edit-find-replace"`

edit find replace

6.38.2.62 `#define EWL_ICON_EDIT_PASTE "edit-paste"`

edit paste

6.38.2.63 `#define EWL_ICON_EDIT_REDO "edit-redo"`

edit redo

6.38.2.64 `#define EWL_ICON_EDIT_SELECT_ALL "edit-select-all"`

edit select all

6.38.2.65 `#define EWL_ICON_EDIT_UNDO "edit-undo"`

edit undo

6.38.2.66 `#define EWL_ICON_EMBLEM_DEFAULT "emblem-default"`

emblem default

6.38.2.67 `#define EWL_ICON_EMBLEM_DOCUMENTS "emblem-documents"`

emblem documents

6.38.2.68 `#define EWL_ICON_EMBLEM_DOWNLOADS "emblem-downloads"`

emblem downloads

6.38.2.69 `#define EWL_ICON_EMBLEM_FAVORITE "emblem-favorite"`

emblem favorite

6.38.2.70 `#define EWL_ICON_EMBLEM_IMPORTANT "emblem-important"`

emblem important

6.38.2.71 `#define EWL_ICON_EMBLEM_MAIL "emblem-mail"`

emblem mail

6.38.2.72 `#define EWL_ICON_EMBLEM_PHOTOS "emblem-photos"`

emblem photos

6.38.2.73 `#define EWL_ICON_EMBLEM_READONLY "emblem-readonly"`

emblem readonly

6.38.2.74 `#define EWL_ICON_EMBLEM_SHARED "emblem-shared"`

emblem shared

6.38.2.75 `#define EWL_ICON_EMBLEM_SYMBOLIC_LINK "emblem-symbolic-link"`

emblem symbolic link

6.38.2.76 `#define EWL_ICON_EMBLEM_SYNCHRONIZED "emblem-synchronized"`

emblem synchronized

6.38.2.77 `#define EWL_ICON_EMBLEM_SYSTEM "emblem-system"`

emblem system

6.38.2.78 `#define EWL_ICON_EMBLEM_UNREADABLE "emblem-unreadable"`

emblem unreadable

6.38.2.79 `#define EWL_ICON_FACE_ANGEL "face-angel"`

face angel

6.38.2.80 `#define EWL_ICON_FACE_CRYING "face-crying"`

face crying

6.38.2.81 `#define EWL_ICON_FACE_DEVIL_GRIN "face-devil-grin"`

face devil grin

6.38.2.82 `#define EWL_ICON_FACE_DEVIL_SAD "face-devil-sad"`

face devil sad

6.38.2.83 `#define EWL_ICON_FACE_GLASSES "face-glasses"`

face glasses

6.38.2.84 `#define EWL_ICON_FACE_KISS "face-kiss"`

face kiss

6.38.2.85 `#define EWL_ICON_FACE_MONKEY "face-monkey"`

face monkey

6.38.2.86 `#define EWL_ICON_FACE_PLAIN "face-plain"`

face plain

6.38.2.87 `#define EWL_ICON_FACE_SAD "face-sad"`

face sad

6.38.2.88 `#define EWL_ICON_FACE_SMILE "face-smile"`

face smile

6.38.2.89 `#define EWL_ICON_FACE_SMILE_BIG "face-smile-big"`

face smile big

6.38.2.90 `#define EWL_ICON_FACE_SMIRK "face-smirk"`

face smirk

6.38.2.91 `#define EWL_ICON_FACE_SURPRISE "face-surprise"`

face surprise

6.38.2.92 `#define EWL_ICON_FACE_WINK "face-wink"`

face wink

6.38.2.93 `#define EWL_ICON_FOLDER "folder"`

folder

6.38.2.94 `#define EWL_ICON_FOLDER_DRAG_ACCEPT "folder-drag-accept"`

folder drag accept

6.38.2.95 `#define EWL_ICON_FOLDER_OPEN "folder-open"`

folder open

6.38.2.96 `#define EWL_ICON_FOLDER_REMOTE "folder-remote"`

folder remote

6.38.2.97 `#define EWL_ICON_FOLDER_VISITING "folder-visiting"`

folder visiting

6.38.2.98 `#define EWL_ICON_FONT_X_GENERIC "font-x-generic"`

font/x-generic

6.38.2.99 `#define EWL_ICON_FORMAT_INDENT_LESS "format-indent-less"`

format indent less

6.38.2.100 `#define EWL_ICON_FORMAT_INDENT_MORE "format-indent-more"`

format indent more

6.38.2.101 `#define EWL_ICON_FORMAT_JUSTIFY_CENTER "format-justify-center"`

format justify center

6.38.2.102 `#define EWL_ICON_FORMAT_JUSTIFY_FILL "format-justify-fill"`

format justify fill

6.38.2.103 `#define EWL_ICON_FORMAT_JUSTIFY_LEFT "format-justify-left"`

format justify left

6.38.2.104 `#define EWL_ICON_FORMAT_JUSTIFY_RIGHT "format-justify-right"`

format justify right

6.38.2.105 `#define EWL_ICON_FORMAT_TEXT_BOLD "format-text-bold"`

format text bold

6.38.2.106 `#define EWL_ICON_FORMAT_TEXT_DIRECTION_LTR "format-text-direction-ltr"`

format text direction left to right

6.38.2.107 `#define EWL_ICON_FORMAT_TEXT_DIRECTION _
RTL "format-text-direction-rtl"`

format text direction right to left

6.38.2.108 `#define EWL_ICON_FORMAT_TEXT_ITALIC "format-text-italic"`

format text italic

6.38.2.109 `#define EWL_ICON_FORMAT_TEXT _
STRIKETHROUGH "format-text-strikethrough"`

format text strikethrough

6.38.2.110 `#define EWL_ICON_FORMAT_TEXT_UNDERLINE "format-text-
underline"`

format text underline

6.38.2.111 `#define EWL_ICON_GO_BOTTOM "go-bottom"`

go bottom

6.38.2.112 `#define EWL_ICON_GO_DOWN "go-down"`

go down

6.38.2.113 `#define EWL_ICON_GO_FIRST "go-first"`

go first

6.38.2.114 `#define EWL_ICON_GO_HOME "go-home"`

go home

6.38.2.115 `#define EWL_ICON_GO_JUMP "go-jump"`

go jump

6.38.2.116 `#define EWL_ICON_GO_LAST "go-last"`

go last

6.38.2.117 `#define EWL_ICON_GO_NEXT "go-next"`

go next

6.38.2.118 `#define EWL_ICON_GO_PREVIOUS "go-previous"`

go previous

6.38.2.119 `#define EWL_ICON_GO_TOP "go-top"`

go top

6.38.2.120 `#define EWL_ICON_GO_UP "go-up"`

go up

6.38.2.121 `#define EWL_ICON_HELP_ABOUT "help-about"`

help about

6.38.2.122 `#define EWL_ICON_HELP_BROWSER "help-browser"`

help browser

6.38.2.123 `#define EWL_ICON_HELP_CONTENTS "help-contents"`

help contents

6.38.2.124 `#define EWL_ICON_HELP_FAQ "help-faq"`

help faq

6.38.2.125 `#define EWL_ICON_IMAGE_LOADING "image-loading"`

image loading

6.38.2.126 `#define EWL_ICON_IMAGE_MISSING "image-missing"`

image missing

6.38.2.127 `#define EWL_ICON_IMAGE_X_GENERIC "image-x-generic"`

image/x-generic

6.38.2.128 `#define EWL_ICON_INPUT_GAMING "input-gaming"`

input gaming

6.38.2.129 `#define EWL_ICON_INPUT_KEYBOARD "input-keyboard"`

input keyboard

6.38.2.130 `#define EWL_ICON_INPUT_MOUSE "input-mouse"`

input mouse

6.38.2.131 `#define EWL_ICON_INSERT_IMAGE "insert-image"`

insert image

6.38.2.132 `#define EWL_ICON_INSERT_LINK "insert-link"`

insert link

6.38.2.133 `#define EWL_ICON_INSERT_OBJECT "insert-object"`

insert object

6.38.2.134 `#define EWL_ICON_INSERT_TEXT "insert-text"`

insert text

6.38.2.135 `#define EWL_ICON_LIST_ADD "list-add"`

list add

6.38.2.136 `#define EWL_ICON_LIST_REMOVE "list-remove"`

list remove

6.38.2.137 `#define EWL_ICON_MAIL_ATTACHMENT "mail-attachment"`

mail attachment

6.38.2.138 `#define EWL_ICON_MAIL_FORWARD "mail-forward"`

mail forward

6.38.2.139 `#define EWL_ICON_MAIL_MARK_IMPORTANT "mail-mark-important"`

mail mark important

6.38.2.140 `#define EWL_ICON_MAIL_MARK_JUNK "mail-mark-junk"`

mail mark junk

6.38.2.141 `#define EWL_ICON_MAIL_MARK_NOTJUNK "mail-mark-notjunk"`

mail mark not junk

6.38.2.142 `#define EWL_ICON_MAIL_MARK_READ "mail-mark-read"`

mail mark read

6.38.2.143 `#define EWL_ICON_MAIL_MARK_UNREAD "mail-mark-unread"`

mail mark unread

6.38.2.144 `#define EWL_ICON_MAIL_MESSAGE_NEW "mail-message-new"`

mail message new

6.38.2.145 `#define EWL_ICON_MAIL_READ "mail-read"`

mail read

6.38.2.146 `#define EWL_ICON_MAIL_REPLIED "mail-replied"`

mail replied

6.38.2.147 `#define EWL_ICON_MAIL_REPLY_ALL "mail-reply-all"`

mail reply all

6.38.2.148 `#define EWL_ICON_MAIL_REPLY_SENDER "mail-reply-sender"`

mail reply sender

6.38.2.149 `#define EWL_ICON_MAIL_SEND_RECEIVE "mail-send-receive"`

mail send receive

6.38.2.150 `#define EWL_ICON_MAIL_SIGNED "mail-signed"`

mail signed

6.38.2.151 `#define EWL_ICON_MAIL_SIGNED_VERIFIED "mail-signed-verified"`

mail signed verified

6.38.2.152 `#define EWL _ICON _MAIL _UNREAD "mail-unread"`

mail unread

6.38.2.153 `#define EWL _ICON _MEDIA _CDROM "media-cdrom"`

media cdrom

6.38.2.154 `#define EWL _ICON _MEDIA _EJECT "media-eject"`

media eject

6.38.2.155 `#define EWL _ICON _MEDIA _FLOPPY "media-floppy"`

media floppy

6.38.2.156 `#define EWL _ICON _MEDIA _PLAYBACK _PAUSE "media-playback-pause"`

media playback pause

6.38.2.157 `#define EWL _ICON _MEDIA _PLAYBACK _START "media-playback-start"`

media playback start

6.38.2.158 `#define EWL _ICON _MEDIA _PLAYBACK _STOP "media-playback-stop"`

media playback stop

6.38.2.159 `#define EWL _ICON _MEDIA _PLAYLIST _REPEAT "media-playlist-repeat"`

media playlist repeat

6.38.2.160 `#define EWL _ICON _MEDIA _PLAYLIST _SHUFFLE "media-playlist-shuffle"`

media playlist shuffle

6.38.2.161 `#define EWL _ICON _MEDIA _RECORD "media-record"`

media record

6.38.2.162 `#define EWL_ICON_MEDIA_SEEK_BACKWARD "media-seek-backward"`

media seek backward

6.38.2.163 `#define EWL_ICON_MEDIA_SEEK_FORWARD "media-seek-forward"`

media seek forward

6.38.2.164 `#define EWL_ICON_MEDIA_SKIP_BACKWARD "media-skip-backward"`

media skip backward

6.38.2.165 `#define EWL_ICON_MEDIA_SKIP_FORWARD "media-skip-forward"`

media skip forward

6.38.2.166 `#define EWL_ICON_MULTIMEDIA_PLAYER "multimedia-player"`

multimedia player

6.38.2.167 `#define EWL_ICON_MULTIMEDIA_VOLUME_CONTROL "multimedia-volume-control"`

multimedia volume control

6.38.2.168 `#define EWL_ICON_NETWORK_ERROR "network-error"`

network error

6.38.2.169 `#define EWL_ICON_NETWORK_IDLE "network-idle"`

network idle

6.38.2.170 `#define EWL_ICON_NETWORK_OFFLINE "network-offline"`

network offline

6.38.2.171 `#define EWL_ICON_NETWORK_RECEIVE "network-receive"`

network receive

6.38.2.172 `#define EWL_ICON_NETWORK_SERVER "network-server"`

network server

6.38.2.173 `#define EWL_ICON_NETWORK_TRANSMIT "network-transmit"`

network transmit

6.38.2.174 `#define EWL_ICON_NETWORK_TRANSMIT _
RECEIVE "network-transmit-receive"`

network transmit receive

6.38.2.175 `#define EWL_ICON_NETWORK_WIRED "network-wired"`

network wired

6.38.2.176 `#define EWL_ICON_NETWORK_WIRELESS "network-wireless"`

network wireless

6.38.2.177 `#define EWL_ICON_NETWORK_WORKGROUP "network-
workgroup"`

network workgroup

6.38.2.178 `#define EWL_ICON_PACKAGE_X_GENERIC "package-x-generic"`

package/x-generic

6.38.2.179 `#define EWL_ICON_PREFERENCES_DESKTOP "preferences-
desktop"`

prefs desktop

6.38.2.180 `#define EWL_ICON_PREFERENCES_DESKTOP _
ACCESSIBILITY "preferences-desktop-accessibility"`

pref desktop accessibility

6.38.2.181 `#define EWL_ICON_PREFERENCES_DESKTOP _
ACCESSIBILITY "preferences-desktop-accessibility"`

pref desktop accessibility

6.38.2.182 `#define EWL_ICON_PREFERENCES_DESKTOP _
FONT "preferences-desktop-font"`

pref desktop font

6.38.2.183 `#define EWL_ICON_PREFERENCES_DESKTOP -
KEYBOARD "preferences-desktop-keyboard"`

pref desktop keyboard

6.38.2.184 `#define EWL_ICON_PREFERENCES_DESKTOP -
LOCALE "preferences-desktop-locale"`

pref desktop locale

6.38.2.185 `#define EWL_ICON_PREFERENCES_DESKTOP -
MULTIMEDIA "preferences-desktop-multimedia"`

pref desktop multimedia

6.38.2.186 `#define EWL_ICON_PREFERENCES_DESKTOP -
PERIPHERALS "preferences-desktop-peripherals"`

prefs desktop peripherals

6.38.2.187 `#define EWL_ICON_PREFERENCES_DESKTOP -
PERSONAL "preferences-desktop-personal"`

prefs desktop personal

6.38.2.188 `#define EWL_ICON_PREFERENCES_DESKTOP -
SCREENSAVER "preferences-desktop-screensaver"`

pref desktop screensaver

6.38.2.189 `#define EWL_ICON_PREFERENCES_DESKTOP -
THEME "preferences-desktop-theme"`

pref desktop theme

6.38.2.190 `#define EWL_ICON_PREFERENCES_DESKTOP -
WALLPAPER "preferences-desktop-wallpaper"`

pref desktop wallpaper

6.38.2.191 `#define EWL_ICON_PREFERENCES_OTHER "preferences-other"`

prefs other

6.38.2.192 `#define EWL_ICON_PREFERENCES_SYSTEM "preferences-
system"`

prefs system

6.38.2.193 `#define EWL_ICON_PREFERENCES_SYSTEM -
NETWORK "preferences-system-network"`

prefs system network

6.38.2.194 `#define EWL_ICON_PRINTER "printer"`

printer

6.38.2.195 `#define EWL_ICON_PRINTER_ERROR "printer-error"`

printer error

6.38.2.196 `#define EWL_ICON_PRINTER_PRINTING "printer-printing"`

printer printing

6.38.2.197 `#define EWL_ICON_PROCESS_WORKING "process-working"`

process working

6.38.2.198 `#define EWL_ICON_SIZE_LARGE 24`

24x24 icons

6.38.2.199 `#define EWL_ICON_SIZE_MEDIUM 22`

22x22 icons

6.38.2.200 `#define EWL_ICON_SIZE_SMALL 16`

16x16 icons

6.38.2.201 `#define EWL_ICON_SOFTWARE_UPDATE -
AVAILABLE "software-update-available"`

software update

6.38.2.202 `#define EWL_ICON_SOFTWARE_UPDATE_URGENT "software-
update-urgent"`

software update urgent

6.38.2.203 `#define EWL_ICON_START_HERE "start-here"`

start here

6.38.2.204 `#define EWL_ICON_SYNC_ERROR "sync-error"`

sync error

6.38.2.205 `#define EWL_ICON_SYNC_SYNCHRONIZING "sync-synchronizing"`

sync synchronizing

6.38.2.206 `#define EWL_ICON_SYSTEM_FILE_MANAGER "system-file-manager"`

system file manager

6.38.2.207 `#define EWL_ICON_SYSTEM_HELP "system-help"`

system help

6.38.2.208 `#define EWL_ICON_SYSTEM_LOCK_SCREEN "system-lock-screen"`

system lock screen

6.38.2.209 `#define EWL_ICON_SYSTEM_LOG_OUT "system-log-out"`

system log out

6.38.2.210 `#define EWL_ICON_SYSTEM_RUN "system-run"`

system run

6.38.2.211 `#define EWL_ICON_SYSTEM_SEARCH "system-search"`

system search

6.38.2.212 `#define EWL_ICON_SYSTEM_SOFTWARE_UPDATE "system-software-update"`

system software update

6.38.2.213 `#define EWL_ICON_TASK_DUE "task-due"`

task due

6.38.2.214 `#define EWL_ICON_TASK_PASSED_DUE "task-passed-due"`

task passwd due

6.38.2.215 `#define EWL _ICON _TEXT _HTML "text-html"`

text/html

6.38.2.216 `#define EWL _ICON _TEXT _X _GENERIC "text-x-generic"`

text/x-generic

6.38.2.217 `#define EWL _ICON _TEXT _X _GENERIC _TEMPLATE "text-x-generic-template"`

text/x-generic-template

6.38.2.218 `#define EWL _ICON _TEXT _X _SCRIPT "text-x-script"`

text/x-script

6.38.2.219 `#define EWL _ICON _TOOLS _CHECK _SPELLING "tools-check-spelling"`

tools check spelling

6.38.2.220 `#define EWL _ICON _USER _AWAY "user-away"`

user away

6.38.2.221 `#define EWL _ICON _USER _DESKTOP "user-desktop"`

user desktop

6.38.2.222 `#define EWL _ICON _USER _HOME "user-home"`

user home

6.38.2.223 `#define EWL _ICON _USER _IDLE "user-idle"`

user idle

6.38.2.224 `#define EWL _ICON _USER _OFFLINE "user-offline"`

user offline

6.38.2.225 `#define EWL _ICON _USER _ONLINE "user-online"`

user online

6.38.2.226 `#define EWL_ICON_USER_TRASH "user-trash"`

user trash

6.38.2.227 `#define EWL_ICON_USER_TRASH_FULL "user-trash-full"`

user trash full

6.38.2.228 `#define EWL_ICON_UTILITIES_TERMINAL "utilities-terminal"`

utilities terminal

6.38.2.229 `#define EWL_ICON_VIDEO_X_GENERIC "video-x-generic"`

video/x-generic

6.38.2.230 `#define EWL_ICON_VIEW_FULLSCREEN "view-fullscreen"`

view fullscreen

6.38.2.231 `#define EWL_ICON_VIEW_REFRESH "view-refresh"`

view refresh

6.38.2.232 `#define EWL_ICON_VIEW_SORT_ASCENDING "view-sort-ascending"`

view sort assending

6.38.2.233 `#define EWL_ICON_VIEW_SORT_DESCENDING "view-sort-descending"`

view sort descending

6.38.2.234 `#define EWL_ICON_WEATHER_CLEAR "weather-clear"`

weather clear

6.38.2.235 `#define EWL_ICON_WEATHER_CLEAR_NIGHT "weather-clear-night"`

weather clear night

6.38.2.236 `#define EWL_ICON_WEATHER_FEW_CLOUDS "weather-few-clouds"`

weather few clouds


```
6.38.2.237 #define EWL_ICON_WEATHER_FEW_CLOUDS -  
          NIGHT "weather-few-clouds-night"
```

weather few clouds night

```
6.38.2.238 #define EWL_ICON_WEATHER_FOG "weather-fog"
```

weather fog

```
6.38.2.239 #define EWL_ICON_WEATHER_OVERCAST "weather-overcast"
```

weather overcast

```
6.38.2.240 #define EWL_ICON_WEATHER_SEVERE_ALERT "weather-  
          severe-alert"
```

weather severe alert

```
6.38.2.241 #define EWL_ICON_WEATHER_SHOWERS "weather-showers"
```

weather showers

```
6.38.2.242 #define EWL_ICON_WEATHER_SHOWERS -  
          SCATTERED "weather-showers-scattered"
```

weather showers scattered

```
6.38.2.243 #define EWL_ICON_WEATHER_SNOW "weather-snow"
```

weather snow

```
6.38.2.244 #define EWL_ICON_WEATHER_STORM "weather-storm"
```

weather storm

```
6.38.2.245 #define EWL_ICON_WINDOW_CLOSE "window-close"
```

window close

```
6.38.2.246 #define EWL_ICON_WINDOW_NEW "window-new"
```

window new

```
6.38.2.247 #define EWL_ICON_X_OFFICE_ADDRESS_BOOK "x-office-  
          address-book"
```

x-office/address-book

6.38.2.248 `#define EWL_ICON_X_OFFICE_CALENDAR "x-office-calendar"`

x-office/calendar

6.38.2.249 `#define EWL_ICON_X_OFFICE_DOCUMENT "x-office-document"`

x-office/document

6.38.2.250 `#define EWL_ICON_X_OFFICE_PRESENTATION "x-office-presentation"`

x-office/presentation

6.38.2.251 `#define EWL_ICON_X_OFFICE_SPREADSHEET "x-office-spreadsheet"`

x-office/spreadsheet

6.38.2.252 `#define EWL_ICON_ZOOM_BEST_FIT "zoom-best-fit"`

zoom best fit

6.38.2.253 `#define EWL_ICON_ZOOM_IN "zoom-in"`

zoom in

6.38.2.254 `#define EWL_ICON_ZOOM_ORIGINAL "zoom-original"`

zoom original

6.38.2.255 `#define EWL_ICON_ZOOM_OUT "zoom-out"`

zoom out

6.38.3 Function Documentation

6.38.3.1 `const char* ewl_icon_theme_icon_path_get (const char * icon, int size)`

Retrives the full path to the specified icon, or NULL if none found.

Parameters:

icon,: The Icon Spec icon name to lookup

size,: The size of the icon to retrieve. A 0 value will cause the default size to be used.

Returns:

Returns the path to the icon we are looking for or NULL if none found

6.38.3.2 int ewl_icon_theme_init (void)

Initializes the icon theme system.

Returns:

Returns TRUE on success or FALSE on failure

6.38.3.3 void ewl_icon_theme_shutdown (void)

Shuts down the icon theme system.

Returns:

Returns no value.

6.38.3.4 void ewl_icon_theme_theme_change (void)

Called when the icon theme is changed so we can clean up any caching we have in place.

Returns:

Returns no value

6.39 Ewl_Image: An Image Display Widget

6.39.1 Detailed Description

Provides a widget for displaying evas loadable images, and edges.

Remarks:

Inherits from **Ewl_Widget** (p. 785).

6.39.2 Tutorial

The **Ewl_Image** (p. 686) widget provides methods for displaying an image in an EWL application. The image widget also supports a limited set of methods for manipulating the displayed image.

```
Ewl_Widget *image;
image = ewl_image_new();
ewl_image_file_path_set(EWL_IMAGE(image), "/usr/share/ewl/image.png");
ewl_widget_show(image);
```

After creating an image widget, the most common operation is to set the path or key on the widget. The key is only used if an edge is loaded as the image, indicating the group to load. This will cause the widget to setup its sizing information when it is realized based on the original image size. It is also possible to request the current file and key set on the image widget.

```
void ewl_image_file_set(Ewl_Image *i, const char *im, const char *key);
void ewl_image_file_path_set(Ewl_Image *i, const char *im);
const char *ewl_image_file_path_get(Ewl_Image *i);
void ewl_image_file_key_set(Ewl_Image *i, const char *key);
const char *ewl_image_file_key_get(Ewl_Image *i);
```

Along with the image to be displayed, the basic operations of scaling and tiling can be applied to the displayed image. Scaling can be applied on either a relative or absolute scale. The functions that refer to scale are a multiplier of the images set size, so 2.0 would result in an image double the original size. The size functions force the image to be scaled to specified dimensions. Setting a tile on an image causes the image to repeat when the specified dimensions are reached, the x and y coordinates indicate the starting offset within the available area where tiling begins.

```
void ewl_image_scale_set(Ewl_Image *i, double wp, double hp);
void ewl_image_scale_get(Ewl_Image *i, double *wp, double *hp);
void ewl_image_size_set(Ewl_Image *i, int w, int h);
void ewl_image_size_get(Ewl_Image *i, int *w, int *h);
void ewl_image_tile_set(Ewl_Image *i, int x, int y, int w, int h);
```

Two settings also impact the sizing, but are not explicit in how they determine the new size. The proportional setting causes the image to be displayed at the original aspect ratio of the specified image, regardless of scaling. The constrain setting sets a maximum size on the dimensions of an image, but will not scale the image if it's smaller than that size.

```
void ewl_image_proportional_set(Ewl_Image *i, char p);
char ewl_image_proportional_get(Ewl_Image *i);
void ewl_image_constrain_set(Ewl_Image *i, unsigned int size);
unsigned int ewl_image_constrain_get(Ewl_Image *i);
```

Since the image is inherited from the base widget class, all of the standard widget and object operations can be applied to the image widget. This includes resizing, theming, color tinting and packing in containers.

Data Structures

- struct **Ewl_Image**
*Inherits from **Ewl_Widget** (p. 785) and extends to provide an image widget.*
- struct **Ewl_Image_Thumbnail**
*Inherits from **Ewl_Image** (p. 686) and extends to provide reference to original image.*

Defines

- `#define EWL_IMAGE(image) ((Ewl_Image *) image)`
- `#define EWL_IMAGE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_IMAGE_TYPE))`
- `#define EWL_IMAGE_THUMBNAIL(image) ((Ewl_Image_Thumbnail *) image)`
- `#define EWL_IMAGE_THUMBNAIL_TYPE "image_thumbnail"`
- `#define EWL_IMAGE_TYPE "image"`

Typedefs

- `typedef Ewl_Image Ewl_Image`
- `typedef Ewl_Image_Thumbnail Ewl_Image_Thumbnail`

Functions

- `void ewl_image_cb_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_image_cb_destroy (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_image_cb_mouse_down (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_image_cb_mouse_move (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_image_cb_mouse_up (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_image_cb_obscure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_image_cb_reveal (Ewl_Widget *w, void *ev_data, void *user_data)`
- `unsigned int ewl_image_constrain_get (Ewl_Image *i)`
Get the current constrain value set on the image.
- `void ewl_image_constrain_set (Ewl_Image *i, unsigned int size)`
Set a size which, if the image is bigger than, scale proportionally.
- `const char * ewl_image_file_key_get (Ewl_Image *i)`
- `void ewl_image_file_key_set (Ewl_Image *i, const char *key)`
Sets the key to use for the image.
- `const char * ewl_image_file_path_get (Ewl_Image *i)`
get the filename this image uses
- `void ewl_image_file_path_set (Ewl_Image *i, const char *im)`

Sets the path to the image file.

- void **ewl_image_file_set** (**Ewl_Image** *i, const char *im, const char *key)
Change the image file displayed by an image widget.
- void **ewl_image_flip** (**Ewl_Image** *img, **Ewl_Orientation** orient)
Flips the given image in the given direction.
- int **ewl_image_init** (**Ewl_Image** *i)
Initialize an image widget to default values and callbacks.
- **Ewl_Widget** * **ewl_image_new** (void)
Load an image widget with specified image contents.
- char **ewl_image_proportional_get** (**Ewl_Image** *i)
Get boolean to determine how to scale.
- void **ewl_image_proportional_set** (**Ewl_Image** *i, char p)
Set boolean to determine how to scale.
- void **ewl_image_rotate** (**Ewl_Image** *i, **Ewl_Rotate** rotate)
Rotates the given image by the given rotate value.
- void **ewl_image_scale_get** (**Ewl_Image** *i, double *wp, double *hp)
Retrieve the percentage an image is scaled.
- void **ewl_image_scale_set** (**Ewl_Image** *i, double wp, double hp)
Scale image dimensions by a percentage.
- void **ewl_image_size_get** (**Ewl_Image** *i, int *w, int *h)
Scale image dimensions to a specific size.
- void **ewl_image_size_set** (**Ewl_Image** *i, int w, int h)
Scale image dimensions to a specific size.
- **Ewl_Widget** * **ewl_image_thumbnail_get** (**Ewl_Image** *i)
Create a widget representing a thumbnailed version of the image.
- int **ewl_image_thumbnail_init** (**Ewl_Image_Thumbnail** *image)
*Initialize an **Ewl_Image_Thumbnail** (p. 689) to default values.*
- **Ewl_Widget** * **ewl_image_thumbnail_new** (void)
*This will create and initialize a new **Ewl_Image_Thumbnail** (p. 689) widget.*
- void **ewl_image_thumbnail_request** (**Ewl_Image_Thumbnail** *thumb, const char *path)
This will request the generation of a thumbnail for the image given by path.
- void **ewl_image_tile_set** (**Ewl_Image** *i, int x, int y, int w, int h)
Tile the image with the given start position and given size.

- **Ewl_View * ewl_image_view_get** (void)

*Creates and returns a view to be used by **Ewl_Image** (p. 686) widgets.*

6.39.3 Define Documentation

6.39.3.1 **#define EWL_IMAGE(image) ((Ewl_Image *) image)**

Typecase a pointer to an **Ewl_Image** (p. 686) widget

6.39.3.2 **#define EWL_IMAGE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_IMAGE_TYPE))**

Returns TRUE if the widget is an **Ewl_Image** (p. 686), FALSE otherwise

6.39.3.3 **#define EWL_IMAGE_THUMBNAIL(image) ((Ewl_Image_Thumbnail *) image)**

Typecase a pointer to an **Ewl_Image_Thumbnail** (p. 689) widget

6.39.3.4 **#define EWL_IMAGE_THUMBNAIL_TYPE "image_thumbnail"**

The type name for the **Ewl_Image_Thumbnail** (p. 689) widget

6.39.3.5 **#define EWL_IMAGE_TYPE "image"**

The type name for the **Ewl_Image** (p. 686) widget

6.39.4 Typedef Documentation

6.39.4.1 **typedef struct Ewl_Image Ewl_Image**

The **Ewl_Image** (p. 686) widget

6.39.4.2 **typedef struct Ewl_Image_Thumbnail Ewl_Image_Thumbnail**

The **Ewl_Image_Thumbnail** (p. 689) widget

6.39.5 Function Documentation

6.39.5.1 void ewl_image_cb_configure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.39.5.2 void ewl_image_cb_destroy (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.39.5.3 void ewl_image_cb_mouse_down (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.39.5.4 void ewl_image_cb_mouse_move (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.39.5.5 void ewl_image_cb_mouse_up (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.39.5.6 void ewl_image_cb_obscure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.39.5.7 void ewl_image_cb_reveal (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.39.5.8 unsigned int ewl_image_constrain_get (Ewl_Image * *i*)

Get the current constrain value set on the image.

Parameters:

i,: The **Ewl_Image** (p. 686) to get the constrain size from

Returns:

Returns the current constrain size on the image

6.39.5.9 void ewl_image_constrain_set (Ewl_Image * *i*, unsigned int *size*)

Set a size which, if the image is bigger than, scale proportionally.

Parameters:

i,: the image to change constrain setting

size,: the minimum constrain size

Returns:

Returns no value.

Sets a size to scale to proportionally if the image exceeds this size

6.39.5.10 const char* ewl_image_file_key_get (Ewl_Image * *i*)

Parameters:

i,: The image to get the key from

Returns:

Returns the key used for this image

6.39.5.11 void ewl_image_file_key_set (Ewl_Image * *i*, const char * *key*)

Sets the key to use for the image.

Parameters:

i,: The image to set the key into

key,: The key to set into the image

Returns:

Returns no value.

6.39.5.12 const char* ewl_image_file_path_get (Ewl_Image * *i*)

get the filename this image uses

Parameters:

i,: the image widget to get the file of

Returns:

Returns the currently set filename

6.39.5.13 void ewl_image_file_path_set (Ewl_Image * *i*, const char * *path*)

Sets the path to the image file.

Parameters:

i,: The image to set the file into

path,: The path to set into the image

Returns:

Returns no value.

6.39.5.14 void ewl_image_file_set (Ewl_Image * *i*, const char * *im*, const char * *key*)

Change the image file displayed by an image widget.

Parameters:

i,: the image widget to change the displayed image

im,: the path to the new image to be displayed by *i*

key,: the key in the file for the image

Returns:

Returns no value.

Set the image displayed by *i* to the one found at the path *im*. If an edge is used, a minimum size should be specified in the edge or the code.

6.39.5.15 void ewl_image_flip (Ewl_Image * *img*, Ewl_Orientation *orient*)

Flips the given image in the given direction.

Parameters:

img,: The image to flip

orient,: The orientation to flip

Returns:

Returns no value

6.39.5.16 int ewl_image_init (Ewl_Image * *i*)

Initialize an image widget to default values and callbacks.

Parameters:

i,: the image widget to initialize

Returns:

Returns no value.

Sets the fields and callbacks of *i* to their default values.

6.39.5.17 Ewl_Widget* ewl_image_new (void)

Load an image widget with specified image contents.

Returns:

Returns a pointer to a new image widget on success, NULL on failure.

The *k* parameter is primarily used for loading edge groups or keyed data in an image.

6.39.5.18 char ewl_image_proportional_get (Ewl_Image * *i*)

Get boolean to determine how to scale.

Parameters:

i,: the image to get the proportional setting value

Returns:

Returns TRUE if the image is forced proportional, FALSE otherwise.

6.39.5.19 void ewl_image_proportional_set (Ewl_Image * *i*, char *p*)

Set boolean to determine how to scale.

Parameters:

- i*,: the image to change proportional setting
- p*,: the boolean indicator of proportionality

Returns:

Returns no value.

Changes the flag indicating if the image is scaled proportionally.

6.39.5.20 void ewl_image_rotate (Ewl_Image * *i*, Ewl_Rotate *rotate*)

Rotates the given image by the given *rotate* value.

Parameters:

- i*,: The image to rotate
- rotate*,: The amount to rotate the image

Returns:

Returns no value

6.39.5.21 void ewl_image_scale_get (Ewl_Image * *i*, double * *wp*, double * *hp*)

Retrieve the percentage an image is scaled.

Parameters:

- i*,: the image to retrieve the current scale
- wp*,: stores the percentage to scale width
- hp*,: stores the percentage to scale height

Returns:

Returns no value.

6.39.5.22 void ewl_image_scale_set (Ewl_Image * *i*, double *wp*, double *hp*)

Scale image dimensions by a percentage.

Parameters:

- i*,: the image to scale
- wp*,: the percentage to scale width
- hp*,: the percentage to scale height

Returns:

Returns no value. Scales the given image to *wp* percent of preferred width by *hp* percent of preferred height. If *i->proportional* is set to TRUE, the lesser of *wp* and *hp* is applied for both directions.

6.39.5.23 void ewl_image_size_get (Ewl_Image * *i*, int * *w*, int * *h*)

Scale image dimensions to a specific size.

Parameters:

i,: the image to scale
w,: the size to scale width
h,: the size to scale height

Returns:

Returns no value.

Scales the given image to *w* by *hp*. If *i->proportional* is set to TRUE, the image is scaled proportional to the lesser scale percentage of preferred size.

6.39.5.24 void ewl_image_size_set (Ewl_Image * *i*, int *w*, int *h*)

Scale image dimensions to a specific size.

Parameters:

i,: the image to scale
w,: the size to scale width
h,: the size to scale height

Returns:

Returns no value.

Scales the given image to *w* by *hp*. If *i->proportional* is set to TRUE, the image is scaled proportional to the lesser scale percentage of preferred size.

6.39.5.25 Ewl_Widget* ewl_image_thumbnail_get (Ewl_Image * *i*)

Create a widget representing a thumbnailed version of the image.

Parameters:

i,: the image to thumbnail

Returns:

Returns a thumbnailed image widget on success, NULL on failure.

6.39.5.26 `int ewl_image_thumbnail_init (Ewl_Image_Thumbnail * image)`

Initialize an **Ewl_Image_Thumbnail** (p. 689) to default values.

Parameters:

image,: The **Ewl_Image_Thumbnail** (p. 689) to init

Returns:

Returns TRUE if successfully initialized, FALSE otherwise

6.39.5.27 `Ewl_Widget* ewl_image_thumbnail_new (void)`

This will create and initialize a new **Ewl_Image_Thumbnail** (p. 689) widget.

Returns:

Returns a new **Ewl_Image_Thumbnail** (p. 689) widget

6.39.5.28 `void ewl_image_thumbnail_request (Ewl_Image_Thumbnail * thumb,
const char * path)`

This will request the generation of a thumbnail for the image given by *path*.

Parameters:

thumb,: The thumbnail to request

path,: The path to the image

Returns:

Returns no value.

6.39.5.29 `void ewl_image_tile_set (Ewl_Image * i, int x, int y, int w, int h)`

Tile the image with the given start position and given size.

Parameters:

i,: the image to tile

x,: the x position of the top right corner

y,: the y position of the top right corner

w,: the width of the tile

h,: the height of the tile

Returns:

Returns no value

Tiles the image across the available area, starting the image at the given position and with the given size.

6.39.5.30 Ewl_View* ewl_image_view_get (void)

Creates and returns a view to be used by **Ewl_Image** (p. 686) widgets.

Returns:

Returns a view that can be used to display **Ewl_Image** (p. 686) widgets

6.40 Ewl_Label: A fully-themable text label

6.40.1 Detailed Description

Defines a class for a fully-themable text label

Remarks:

Inherits from **Ewl_Widget** (p. 785).

Data Structures

- struct **Ewl_Label**
*Inherits from **Ewl_Widget** (p. 785) class to provide a simple label.*

Defines

- #define **EWL_LABEL**(label) ((**Ewl_Label** *)label)
- #define **EWL_LABEL_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_LABEL_TYPE**))
- #define **EWL_LABEL_TYPE** "label"

Typedefs

- typedef **Ewl_Label** **Ewl_Label**

Functions

- int **ewl_label_init** (**Ewl_Label** *la)
Initializes the la widget.
- **Ewl_Widget** * **ewl_label_new** (void)
*Creates a new **Ewl_Label** (p. 692) widget with the text text in it.*
- const char * **ewl_label_text_get** (**Ewl_Label** *la)
*Gets the current text set into the label. Note, this pointer will only be valid until the next time **ewl_label_text_set**() (p. 289) is called.*
- void **ewl_label_text_set** (**Ewl_Label** *la, const char *text)
Sets the given text into the widget la.
- **Ewl_View** * **ewl_label_view_get** (void)
*Creates and returns a view to be used by **Ewl_Label** (p. 692) widgets.*

6.40.2 Define Documentation

6.40.2.1 #define **EWL_LABEL**(label) ((**Ewl_Label** *)label)

Typecasts a pointer to an **Ewl_Label** (p. 692) pointer

6.40.2.2 `#define EWL_LABEL_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_LABEL_TYPE))`

Returns TRUE if the widget is an **Ewl_Label** (p. 692), FALSE otherwise

6.40.2.3 `#define EWL_LABEL_TYPE "label"`

The type name for the **Ewl_Label** (p. 692) widget

6.40.3 Typedef Documentation

6.40.3.1 `typedef struct Ewl_Label Ewl_Label`

The **Ewl_Label** (p. 692) widget

6.40.4 Function Documentation

6.40.4.1 `int ewl_label_init (Ewl_Label * la)`

Initializes the *la* widget.

Parameters:

la,: The **Ewl_Label** (p. 692) to initialize

Returns:

Returns TRUE on success, FALSE on failure

6.40.4.2 `Ewl_Widget* ewl_label_new (void)`

Creates a new **Ewl_Label** (p. 692) widget with the *text* text in it.

Returns:

Returns a new **Ewl_Widget** (p. 785) if successful, NULL on failure

6.40.4.3 `const char* ewl_label_text_get (Ewl_Label * la)`

Gets the current text set into the label. Note, this pointer will only be valid until the next time `ewl_label_text_set()` (p. 289) is called.

Parameters:

la,: The **Ewl_Label** (p. 692) to get the text from

Returns:

Returns no value.

6.40.4.4 void ewl_label_text_set (Ewl_Label * *la*, const char * *text*)

Sets the given *text* into the widget *la*.

Parameters:

la,: The **Ewl_Widget** (p. 785) to set the text on

text,: The text to set into the widget

Returns:

Returns no value

6.40.4.5 Ewl_View* ewl_label_view_get (void)

Creates and returns a view to be used by **Ewl_Label** (p. 692) widgets.

Returns:

Returns a view that can be used to display **Ewl_Label** (p. 692) widgets

6.41 Ewl_List: A list widget

6.41.1 Detailed Description

Defines a class to layout data in a list.

Remarks:

Inherits from **Ewl_MVC** (p. 705).

Data Structures

- struct **Ewl_List**

*Inherits from **Ewl_MVC** (p. 705) and extends to provide a list widget.*

Defines

- `#define EWL_LIST(list) ((Ewl_List *)list)`
- `#define EWL_LIST_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_LIST_TYPE))`
- `#define EWL_LIST_TYPE "list"`

Typedefs

- `typedef Ewl_List Ewl_List`

Functions

- `void ewl_list_cb_child_add (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_list_cb_configure (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_list_cb_item_clicked (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_list_cb_selected_change (Ewl_MVC *mvc)`
- `int ewl_list_init (Ewl_List *list)`

*Initialises an **Ewl_List** (p. 693) widget to default values.*

- `Ewl_Widget * ewl_list_new (void)`

*Creates and initializes a new **Ewl_List** (p. 693) widget.*

6.41.2 Define Documentation

6.41.2.1 `#define EWL_LIST(list) ((Ewl_List *)list)`

Typecasts a pointer to an **Ewl_List** (p. 693) pointer

6.41.2.2 `#define EWL_LIST_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_LIST_TYPE))`

Returns TRUE if the widget is an **Ewl_List**, FALSE otherwise

6.41.2.3 `#define EWL_LIST_TYPE "list"`

The type name for the **Ewl_List** (p. 693) widget

6.41.3 Typedef Documentation

6.41.3.1 `typedef struct Ewl_List Ewl_List`

The list structure

6.41.4 Function Documentation

6.41.4.1 `void ewl_list_cb_child_add (Ewl_Container * c, Ewl_Widget * w)`

6.41.4.2 `void ewl_list_cb_configure (Ewl_Widget * w, void * ev, void * data)`

6.41.4.3 `void ewl_list_cb_item_clicked (Ewl_Widget * w, void * ev, void * data)`

6.41.4.4 `void ewl_list_cb_selected_change (Ewl_MVC * mvc)`

6.41.4.5 `int ewl_list_init (Ewl_List * list)`

Initialises an **Ewl_List** (p. 693) widget to default values.

Parameters:

list,: The list to initialize

Returns:

Returns TRUE on success or FALSE on failure

6.41.4.6 `Ewl_Widget* ewl_list_new (void)`

Creates and initializes a new **Ewl_List** (p. 693) widget.

Returns:

Returns a new **Ewl_Widget** (p. 785) on success or NULL on failure

6.42 Ewl_Macros: Useful Macros Used Internally and Available Externally

6.42.1 Detailed Description

Defines a variety of utility macros.

Defines

- `#define FREE(dat)`
- `#define IF_FREE(dat)`
- `#define IF_FREE_HASH(hash)`
- `#define IF_FREE_LIST(list)`
- `#define IF_RELEASE(item)`
- `#define MAX(x, y) (((x) > (y)) ? (x) : (y))`
- `#define MIN(x, y) (((x) > (y)) ? (y) : (x))`
- `#define NEW(type, num) calloc(num, sizeof(type));`
- `#define REALLOC(dat, type, num)`
- `#define ZERO(ptr, type, num) ptr = memset(ptr, 0, sizeof(type) * (num))`

6.42.2 Define Documentation

6.42.2.1 `#define FREE(dat)`

Value:

```
{ \
    free(dat); dat = NULL; \
}
```

Free the data pointed to by *dat* and it to NULL.

6.42.2.2 `#define IF_FREE(dat)`

Value:

```
{ \
    if (dat) FREE(dat); \
}
```

If *dat* is non-NULL, free *dat* and assign it to NULL.

6.42.2.3 `#define IF_FREE_HASH(hash)`

Value:

```
{ \
    if (hash) { \
        ecore_hash_destroy(hash); \
        hash = NULL; \
    } \
}
```

If *hash* is non-NULL, free *hash* and set it NULL.

6.42.2.4 #define IF_FREE_LIST(list)

Value:

```
{ \
    if (list) { \
        ecore_list_destroy(list); \
        list = NULL; \
    } \
}
```

If *list* is non-NULL, free *list* and set it NULL.

6.42.2.5 #define IF_RELEASE(item)

Value:

```
{ \
    if (item) { \
        ecore_string_release(item); \
        item = NULL; \
    } \
}
```

If *item* is non-NULL, `ecore_string_release` *item* and set it NULL.

6.42.2.6 #define MAX(x, y) (((x) > (y)) ? (x) : (y))

Select the maximum of two values

6.42.2.7 #define MIN(x, y) (((x) > (y)) ? (y) : (x))

Select the minimum of two values

6.42.2.8 #define NEW(type, num) calloc(num, sizeof(type));

Allocates memory of *num* elements of `sizeof(type)`.

6.42.2.9 #define REALLOC(dat, type, num)

Value:

```
{ \
    if (dat) \
    { \
        dat = realloc(dat, sizeof(type) * num); \
    } \
}
```

Reallocates memory pointed to by *dat* to *num* elements of `sizeof(type)`.

6.42.2.10 `#define ZERO(ptr, type, num) ptr = memset(ptr, 0, sizeof(type) * (num))`

Set the first *num* elements of `sizeof(type)` pointed to by *ptr* to zero.

6.43 Ewl_Media: A Multi media widget

6.43.1 Detailed Description

Defines a class for displaing video.

Remarks:

Inherits from **Ewl_Widget** (p. 785).

Data Structures

- struct **Ewl_Media**

*Inherits from the **Ewl_Widget** (p. 785) class and extends it to provide for multi-line media layout, obstacle wrapping, and a variety of formatting.*

Defines

- #define **EWL_MEDIA**(media) ((**Ewl_Media** *) media)
- #define **EWL_MEDIA_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_MEDIA_TYPE**))
- #define **EWL_MEDIA_TYPE** "media"

Typedefs

- typedef **Ewl_Media** **Ewl_Media**

Functions

- int **ewl_media_audio_mute_get** (**Ewl_Media** *m)
Checks if the media widget is muted.
- void **ewl_media_audio_mute_set** (**Ewl_Media** *m, int mute)
Mutes the media widget.
- double **ewl_media_audio_volume_get** (**Ewl_Media** *m)
Gets the current volume from the media widget.
- void **ewl_media_audio_volume_set** (**Ewl_Media** *m, double v)
Sets the media widget to the given volume.
- void **ewl_media_cb_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_media_cb_realize** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_media_cb_reveal** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_media_cb_unrealize** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- int **ewl_media_init** (**Ewl_Media** *m)
Initialize the fields and callbacks of a media object.

- unsigned int **ewl_media_is_available** (void)
This will check if the modules required for the media widget were available at compile time or not.
- int **ewl_media_length_get** (Ewl_Media *m)
Retrieve the length of the media displayed by the media widget.
- void **ewl_media_length_time_get** (Ewl_Media *m, int *h, int *min, double *s)
Puts the length of the video into the h, m, s variables.
- const char * **ewl_media_media_get** (Ewl_Media *m)
Retrieve the media of a media widget.
- void **ewl_media_media_set** (Ewl_Media *m, const char *media)
Set the media of a media widget.
- Ewl_Media_Module_Type **ewl_media_module_get** (Ewl_Media *m)
Retrieve the module of a media widget.
- int **ewl_media_module_set** (Ewl_Media *m, Ewl_Media_Module_Type module)
Set the module of a media widget.
- Ewl_Widget * **ewl_media_new** (void)
Allocate a new media widget.
- void **ewl_media_play_set** (Ewl_Media *m, int p)
Sets the media widget into the given state.
- double **ewl_media_position_get** (Ewl_Media *m)
Returns the position of the current media.
- void **ewl_media_position_set** (Ewl_Media *m, double p)
Sets the media widget to the specified position.
- void **ewl_media_position_time_get** (Ewl_Media *m, int *h, int *min, double *s)
Puts the position of the video into the h, m, s variables.
- int **ewl_media_seekable_get** (Ewl_Media *m)
Returns if the media area is seekable.

6.43.2 Define Documentation

6.43.2.1 #define EWL_MEDIA(media) ((Ewl_Media *) media)

Typecasts a pointer to an **Ewl_Media** (p.694) pointer.

6.43.2.2 `#define EWL_MEDIA_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_MEDIA_TYPE))`

Returns TRUE if the widget is an **Ewl_Media** (p. 694), FALSE otherwise

6.43.2.3 `#define EWL_MEDIA_TYPE "media"`

The type name for the **Ewl_Media** (p. 694) widget

6.43.3 Typedef Documentation

6.43.3.1 `typedef struct Ewl_Media Ewl_Media`

The **Ewl_Media** (p. 694) widget

6.43.4 Function Documentation

6.43.4.1 `int ewl_media_audio_mute_get (Ewl_Media * m)`

Checks if the media widget is muted.

Parameters:

m,: the media widget to act upon

Returns:

Returns if the media widget is muted

6.43.4.2 `void ewl_media_audio_mute_set (Ewl_Media * m, int mute)`

Mutes the media widget.

Parameters:

m,: the media widget to act upon

mute,: boolean to indicate if the sound should be muted

Returns:

Returns no value

6.43.4.3 `double ewl_media_audio_volume_get (Ewl_Media * m)`

Gets the current volume from the media widget.

Parameters:

m,: the media widget to act upon

Returns:

Returns the media widget volume

6.43.4.4 void ewl_media_audio_volume_set (Ewl_Media * *m*, double *v*)

Sets the media widget to the given volume.

Parameters:

- m*,: the media widget to act upon
- v*,: the volume to set the widget too

Returns:

Returns no value

6.43.4.5 void ewl_media_cb_configure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)**6.43.4.6 void ewl_media_cb_realize (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)****6.43.4.7 void ewl_media_cb_reveal (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)****6.43.4.8 void ewl_media_cb_unrealize (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)****6.43.4.9 int ewl_media_init (Ewl_Media * *m*)**

Initialize the fields and callbacks of a media object.

Parameters:

- m*,: the media area to be initialized

Returns:

Returns TRUE on success, FALSE on failure.

Sets the internal fields and callbacks of a media object to there defaults.

6.43.4.10 unsigned int ewl_media_is_available (void)

This will check if the modules required for the media widget were available at compile time or not.

Returns:

Returns TRUE if the media widget is available, FALSE otherwise

6.43.4.11 int ewl_media_length_get (Ewl_Media * *m*)

Retrieve the length of the media displayed by the media widget.

Parameters:

m,: the media widget to retrieve length from

Returns:

Returns the length of the media contained in the widget.

6.43.4.12 `void ewl_media_length_time_get (Ewl_Media * m, int * h, int * min, double * s)`

Puts the length of the video into the *h*, *m*, *s* variables.

Parameters:

m,: the media widget to act upon

h,: hours variable

min,: minutes variable

s,: seconds variable

6.43.4.13 `const char* ewl_media_media_get (Ewl_Media * m)`

Retrieve the media of a media widget.

Parameters:

m,: the media widget to retrieve media contents

Returns:

Returns a copy of the media in *m* on success, NULL on failure.

6.43.4.14 `void ewl_media_media_set (Ewl_Media * m, const char * media)`

Set the media of a media widget.

Parameters:

m,: the media area widget to set the media

media,: the media to set in the media widget *m*

Returns:

Returns no value.

Sets the media of the media widget *m*

6.43.4.15 Ewl_Media_Module_Type ewl_media_module_get (Ewl_Media * *m*)

Retrieve the module of a media widget.

Parameters:

m,: the media widget to retrieve module contents

Returns:

Returns a the module associated with the media object

**6.43.4.16 int ewl_media_module_set (Ewl_Media * *m*,
Ewl_Media_Module_Type *module*)**

Set the module of a media widget.

Parameters:

m,: the media area widget to set the module

module,: the module to set in the media widget *m*

Returns:

Returns FALSE if we failed to load the module, TRUE otherwise.

Sets the module of the media widget *m*

6.43.4.17 Ewl_Widget* ewl_media_new (void)

Allocate a new media widget.

Returns:

Returns a pointer to a new media on success, NULL on failure.

6.43.4.18 void ewl_media_play_set (Ewl_Media * *m*, int *p*)

Sets the media widget into the given state.

Parameters:

m,: the media widget to act upon

p,: the value to set play too

Returns:

Returns no value

6.43.4.19 double ewl_media_position_get (Ewl_Media * *m*)

Returns the position of the current media.

Parameters:

m,: the media widget to act upon

Returns:

Returns the current media position

6.43.4.20 void ewl_media_position_set (Ewl_Media * *m*, double *p*)

Sets the media widget to the specified position.

Parameters:

m,: the media widget to act upon

p,: the positon to seek too

Returns:

Returns no value

6.43.4.21 void ewl_media_position_time_get (Ewl_Media * *m*, int * *h*, int * *min*, double * *s*)

Puts the position of the video into the *h*, *m*, *s* variables.

Parameters:

m,: the media widget to act upon

h,: hours variable

min,: minutes variable

s,: seconds variable

6.43.4.22 int ewl_media_seekable_get (Ewl_Media * *m*)

Returns if the media area is seekable.

Parameters:

m,: the media widget to act upon

Returns:

Returns if the media area is seekable

6.44 Ewl_Menu: A Simple Windowed Menu

6.44.1 Detailed Description

Defines a menu used globally. The contents on the menu are drawn outside of the canvas.

Remarks:

Inherits from Ewl_Menu_Base.

Data Structures

- struct **Ewl_Menu**
Inherits from the Ewl_Menu_Base and extends to provide a simple menu.
- struct **Ewl_Menu_Info**
Provides a simple structure to contain menu information.

Defines

- #define **EWL_MENU**(menu) ((Ewl_Menu *) menu)
- #define **EWL_MENU_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_MENU_TYPE))
- #define **EWL_MENU_TYPE** "menu"

Typedefs

- typedef **Ewl_Menu** Ewl_Menu
- typedef **Ewl_Menu_Info** Ewl_Menu_Info

Functions

- void **ewl_menu_cb_configure** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_menu_cb_destroy** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_menu_cb_expand** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_menu_cb_hide** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_menu_cb_mouse_move** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_menu_cb_popup_destroy** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_menu_cb_realize** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_menu_collapse** (Ewl_Menu *menu)
Collapses the popup portion of the menu.
- void **ewl_menu_from_info** (Ewl_Menu *menu, Ewl_Menu_Info *info)
Initializes menu with info.
- int **ewl_menu_init** (Ewl_Menu *menu)
Initialize an internal menu to starting values.

- `int ewl_menu_mouse_feed (Ewl_Menu *menu, int x, int y)`
- `Ewl_Widget * ewl_menu_new (void)`

Create a new internal menu.

6.44.2 Define Documentation

6.44.2.1 `#define EWL_MENU(menu) ((Ewl_Menu *) menu)`

Typecasts a pointer to an **Ewl_Menu** (p. 696) pointer.

6.44.2.2 `#define EWL_MENU_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_MENU_TYPE))`

Returns TRUE if the widget is an **Ewl_Menu** (p. 696), FALSE otherwise

6.44.2.3 `#define EWL_MENU_TYPE "menu"`

The type name for the **Ewl_Menu** (p. 696) widget

6.44.3 Typedef Documentation

6.44.3.1 `typedef struct Ewl_Menu Ewl_Menu`

A simple internal menu, it is limited to drawing within the current evas.

6.44.3.2 `typedef struct Ewl_Menu_Info Ewl_Menu_Info`

A simple struct to contain menu information

6.44.4 Function Documentation

6.44.4.1 void ewl_menu_cb_configure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.44.4.2 void ewl_menu_cb_destroy (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.44.4.3 void ewl_menu_cb_expand (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.44.4.4 void ewl_menu_cb_hide (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.44.4.5 void ewl_menu_cb_mouse_move (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.44.4.6 void ewl_menu_cb_popup_destroy (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.44.4.7 void ewl_menu_cb_realize (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.44.4.8 void ewl_menu_collapse (Ewl_Menu * *menu*)

Collapses the popup portion of the menu.

Parameters:

menu,: the menu to work with

Returns:

Returns no value

6.44.4.9 void ewl_menu_from_info (Ewl_Menu * *menu*, Ewl_Menu_Info * *info*)

Initializes *menu* with *info*.

Parameters:

menu,: The menu to setup

info,: The info to set into the menu

Returns:

Returns no value

6.44.4.10 int ewl_menu_init (Ewl_Menu * *menu*)

Initialize an internal menu to starting values.

Parameters:

menu,: the menu to initialize

Returns:

Returns no value.

6.44.4.11 `int ewl_menu_mouse_feed (Ewl_Menu * menu, int x, int y)`

6.44.4.12 `Ewl_Widget* ewl_menu_new (void)`

Create a new internal menu.

Returns:

Returns a pointer to a new menu on success, NULL on failure.

6.45 Ewl_Menu_Item: The basic menu item

6.45.1 Detailed Description

The basic menu item

Remarks:

Inherits from **Ewl_Button** (p. 589).

Data Structures

- struct **Ewl_Menu_Item**

*Inherits from **Ewl_Button** (p. 589) and provides a basic menu item widget.*

Defines

- `#define EWL_MENU_ITEM(mi) ((Ewl_Menu_Item *) mi)`
- `#define EWL_MENU_ITEM_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_MENU_ITEM_TYPE))`
- `#define EWL_MENU_ITEM_TYPE "menu_item"`

Typedefs

- `typedef Ewl_Menu_Item Ewl_Menu_Item`

Functions

- `int ewl_menu_item_init (Ewl_Menu_Item *menu)`
Initialize the fields of a menu item to their defaults.
- `Ewl_Widget * ewl_menu_item_new (void)`
Create a new menu item to place in a menu.

6.45.2 Define Documentation

6.45.2.1 `#define EWL_MENU_ITEM(mi) ((Ewl_Menu_Item *) mi)`

Typecasts a pointer to an **Ewl_Menu_Item** (p. 699) pointer.

6.45.2.2 `#define EWL_MENU_ITEM_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_MENU_ITEM_TYPE))`

Returns TRUE if the widget is an **Ewl_Menu_Item** (p. 699), FALSE otherwise

6.45.2.3 `#define EWL_MENU_ITEM_TYPE "menu_item"`

The type name for the `Ewl_Menu_Item` (p. 699) widget

6.45.3 Typedef Documentation

6.45.3.1 `typedef struct Ewl_Menu_Item Ewl_Menu_Item`

This serves as a basis for all menu related entries. It provides the most basic layout facilities for items in a menu.

6.45.4 Function Documentation

6.45.4.1 `int ewl_menu_item_init (Ewl_Menu_Item * item)`

Initialize the fields of a menu item to their defaults.

Parameters:

item,: the item to be initialized

Returns:

Returns TRUE on success or FALSE on failure

6.45.4.2 `Ewl_Widget* ewl_menu_item_new (void)`

Create a new menu item to place in a menu.

Returns:

Returns a pointer to a new menu item on success, NULL on failure.

6.46 Ewl_Menubar: A menu bar widget

6.46.1 Detailed Description

Defines a class for displaying menu bars

Remarks:

Inherits from **Ewl_Box** (p. 586).

Data Structures

- struct **Ewl_Menubar**
*Inherits from the **Ewl_Box** (p. 586) class and extends it to provide for a menubar.*
- struct **Ewl_Menubar_Info**
Provides a simple menu structure for the menubar.

Defines

- `#define EWL_MENUBAR(menubar) ((Ewl_Menubar *) menubar)`
- `#define EWL_MENUBAR_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_MENUBAR_TYPE))`
- `#define EWL_MENUBAR_TYPE "menubar"`

Typedefs

- `typedef Ewl_Menubar Ewl_Menubar`
- `typedef Ewl_Menubar_Info Ewl_Menubar_Info`

Functions

- `Ewl_Widget * ewl_hmenubar_new (void)`
*Creates a new **Ewl_Menubar** (p. 701) in a horizontal orientation.*
- `void ewl_menubar_cb_child_add (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_menubar_from_info (Ewl_Menubar *mb, Ewl_Menubar_Info *info)`
Sets up the menubar mb with the given info.
- `int ewl_menubar_init (Ewl_Menubar *mb)`
Initialize a menubar to default values.
- `Ewl_Widget * ewl_menubar_new (void)`
Allocate and initialize a new menubar widget.
- `Ewl_Orientation ewl_menubar_orientation_get (Ewl_Menubar *mb)`
Gets the orientation of the menubar.

- `void ewl__menubar__orientation__set (Ewl_Menubar *mb, Ewl_Orientation o)`
Sets the orientation of the menubar.
- `Ewl_Widget * ewl__vmenubar__new (void)`
*Creates a new **Ewl_Menubar** (p. 701) in a vertical orientation.*

6.46.2 Define Documentation

6.46.2.1 `#define EWL_MENUBAR(menubar) ((Ewl_Menubar *) menubar)`

Typecasts a pointer to a **Ewl_Menubar** (p. 701) pointer

6.46.2.2 `#define EWL_MENUBAR_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_MENUBAR_TYPE))`

Returns TRUE if the widget is an **Ewl_Menubar** (p. 701), FALSE otherwise

6.46.2.3 `#define EWL_MENUBAR_TYPE "menubar"`

The type name for the **Ewl_Menubar** (p. 701) widget

6.46.3 Typedef Documentation

6.46.3.1 `typedef struct Ewl_Menubar Ewl_Menubar`

The **Ewl_Menubar** (p. 701) widget

6.46.3.2 `typedef struct Ewl_Menubar_Info Ewl_Menubar_Info`

Ewl_Menubar_Info (p. 702)

6.46.4 Function Documentation

6.46.4.1 `Ewl_Widget* ewl__hmenubar__new (void)`

Creates a new **Ewl_Menubar** (p. 701) in a horizontal orientation.

Returns:

Returns NULL on failure or a pointer to a new horizontal menubar on success

6.46.4.2 `void ewl__menubar__cb_child__add (Ewl_Container * c, Ewl_Widget * w)`

6.46.4.3 `void ewl__menubar__from__info (Ewl_Menubar * mb, Ewl_Menubar_Info * info)`

Sets up the menubar *mb* with the given *info*.

Parameters:

mb,: The menubar to work with
info,: The menubar info to work with

Returns:

Returns no value

6.46.4.4 int ewl_menubar_init (Ewl_Menubar * *mb*)

Initialize a menubar to default values.

Parameters:

mb,: the menubar to initialize

Returns:

Returns TRUE on success and FALSE on failure

6.46.4.5 Ewl_Widget* ewl_menubar_new (void)

Allocate and initialize a new menubar widget.

Returns:

Returns NULL on failure, or a pointer to a new menubar on success.

6.46.4.6 Ewl_Orientation ewl_menubar_orientation_get (Ewl_Menubar * *mb*)

Gets the orientation of the menubar.

Parameters:

mb,: The menubar to get the orientation from

Returns:

Returns the orientation of the menubar

**6.46.4.7 void ewl_menubar_orientation_set (Ewl_Menubar * *mb*,
Ewl_Orientation *o*)**

Sets the orientation of the menubar.

Parameters:

mb,: The menubar to set the orientation on
o,: The orientation to set onto the menubar

Returns:

Returns no value.

6.46.4.8 Ewl_Widget* ewl_vmenubar_new (void)

Creates a new **Ewl_Menubar** (p. 701) in a vertical orientation.

Returns:

Returns NULL on failure or a pointer to a new vertical menubar on success

6.47 Ewl_Misc: Miscellaneous Utility Functions

6.47.1 Detailed Description

Provides important miscellaneous functionality such as manipulation of the main loop.

Typedefs

- typedef void(*) **Ewl_Shutdown_Hook** (void)

Functions

- void **ewl_canvas_destroy** (void *evas)
- void **ewl_canvas_object_destroy** (void *obj)
- void **ewl_configure_request** (**Ewl_Widget** *w)
Ask for a widget to be configured during idle loop.
- void **ewl_debug_indent_print** (int mod_dir)
Creates a string used to indent debug messages.
- void **ewl_destroy_request** (**Ewl_Widget** *w)
- int **ewl_init** (int *argc, char **argv)
Initialize the internal variables of ewl to begin the program.
- void **ewl_main** (void)
The main execution loop of EWL.
- void **ewl_main_quit** (void)
Notifies ewl to quit at the end of this pass of the main loop.
- void **ewl_print_help** (void)
Print out the Ewl help text.
- void **ewl_realize_cancel_request** (**Ewl_Widget** *w)
- void **ewl_realize_request** (**Ewl_Widget** *w)
- int **ewl_shutdown** (void)
Cleanup internal data structures used by ewl.
- void **ewl_shutdown_add** (**Ewl_Shutdown_Hook** hook)
Allows a widget to hookup a shutdown callback that will be called when Ewl is shutdown.

6.47.2 Typedef Documentation

6.47.2.1 `typedef void(*) Ewl_Shutdown_Hook(void)`

6.47.3 Function Documentation

6.47.3.1 `void ewl_canvas_destroy (void * evas)`

6.47.3.2 `void ewl_canvas_object_destroy (void * obj)`

6.47.3.3 `void ewl_configure_request (Ewl_Widget * w)`

Ask for a widget to be configured during idle loop.

Parameters:

w,: the widget to register for configuration

Returns:

Returns no value.

Ask for the widget *w* to be configured when the main idle loop is executed.

6.47.3.4 `void ewl_debug_indent_print (int mod_dir)`

Creates a string used to indent debug messages.

Parameters:

mod_dir,: do we add or remove from the indent

Returns:

Returns a string with a number of spaces equal to the current debug level

6.47.3.5 `void ewl_destroy_request (Ewl_Widget * w)`

6.47.3.6 `int ewl_init (int * argc, char ** argv)`

Initialize the internal variables of ewl to begin the program.

Parameters:

argc,: the argc passed into the main function

argv,: the argv passed into the main function

Returns:

Returns 1 or greater on success, 0 otherwise.

Sets up necessary internal variables for executing ewl functions. This should be called before any other ewl functions are used.

6.47.3.7 void ewl_main (void)

The main execution loop of EWL.

Returns:

Returns no value.

This is the main execution loop of ewl. It dispatches incoming events and renders updates to the evas's used by ewl.

6.47.3.8 void ewl_main_quit (void)

Notifies ewl to quit at the end of this pass of the main loop.

Returns:

Returns no value.

Sets ewl to exit the main execution loop after this time through the loop has been completed.

6.47.3.9 void ewl_print_help (void)

Print out the Ewl help text.

Returns:

Returns no value

6.47.3.10 void ewl_realize_cancel_request (Ewl_Widget * *w*)**6.47.3.11 void ewl_realize_request (Ewl_Widget * *w*)****6.47.3.12 int ewl_shutdown (void)**

Cleanup internal data structures used by ewl.

This should be called to cleanup internal EWL data structures, if using ecore directly rather than using `ewl_main()` (p. 314).

6.47.3.13 void ewl_shutdown_add (Ewl_Shutdown_Hook *hook*)

Allows a widget to hookup a shutdown callback that will be called when Ewl is shutdown.

Parameters:

hook,: The shutdown callback to call

Returns:

Returns no value

6.48 Ewl_Model: A data model

6.48.1 Detailed Description

Defines communication callbacks for views and controllers. Query row/column data, indicate expansion points, notify views and controllers of changes, trigger sorting on a row/column combination.

Data Structures

- struct **Ewl_Model**

Holds the callbacks needed to define a model.

Defines

- `#define EWL_MODEL(model) ((Ewl_Model *)model)`
- `#define EWL_MODEL_COLUMN_SORTABLE(f) ((Ewl_Model_Column_Sortable)f)`
- `#define EWL_MODEL_DATA_COUNT(f) ((Ewl_Model_Data_Count)f)`
- `#define EWL_MODEL_DATA_EXPANDABLE(f) ((Ewl_Model_Data_Expandable)f)`
- `#define EWL_MODEL_DATA_FETCH(f) ((Ewl_Model_Data_Fetch)f)`
- `#define EWL_MODEL_DATA_HEADER_FETCH(f) ((Ewl_Model_Data_Header_Fetch)f)`
- `#define EWL_MODEL_DATA_SORT(f) ((Ewl_Model_Data_Sort)f)`
- `#define EWL_MODEL_EXPANSION_DATA_FETCH(f) ((Ewl_Model_Expansion_Data_Fetch)f)`
- `#define EWL_MODEL_EXPANSION_MODEL_FETCH(f) ((Ewl_Model_Expansion_Model_Fetch)f)`
- `#define EWL_MODEL_FREE(f) ((Ewl_Model_Data_Free)f)`

Typedefs

- `typedef Ewl_Model Ewl_Model`
- `typedef int(*) Ewl_Model_Column_Sortable (void *data, int col)`
- `typedef int(*) Ewl_Model_Data_Count (void *data)`
- `typedef int(*) Ewl_Model_Data_Expandable (void *data, unsigned int row)`
- `typedef void *(*) Ewl_Model_Data_Fetch (void *data, unsigned int row, unsigned int column)`
- `typedef void(*) Ewl_Model_Data_Free (void *data)`
- `typedef void *(*) Ewl_Model_Data_Header_Fetch (void *data, unsigned int col)`
- `typedef void(*) Ewl_Model_Data_Sort (void *data, unsigned int column, Ewl_Sort_Direction sort)`
- `typedef void *(*) Ewl_Model_Expansion_Data_Fetch (void *data, unsigned int row)`
- `typedef Ewl_Model *(*) Ewl_Model_Expansion_Model_Fetch (void *data, unsigned int row)`

Functions

- **int ewl_model_cb_ecore_list_count** (void *data)
This will return the number of elements in the ecore_list.
- **void * ewl_model_cb_ecore_list_fetch** (void *data, unsigned int row, unsigned int col)
This will return the row data element from the list.
- **Ewl_Model_Column_Sortable ewl_model_column_sortable_get** (Ewl_Model *m)
Retrieves the sortable callback on the model.
- **void ewl_model_column_sortable_set** (Ewl_Model *m, Ewl_Model_Column_Sortable sortable)
Sets the sortable callback.
- **Ewl_Model_Data_Count ewl_model_data_count_get** (Ewl_Model *m)
Gets the count callback from the model.
- **void ewl_model_data_count_set** (Ewl_Model *m, Ewl_Model_Data_Count count)
Sets the count callback into the model.
- **Ewl_Model_Data_Expandable ewl_model_data_expandable_get** (Ewl_Model *m)
Retrieves the expandable callback set on the model.
- **void ewl_model_data_expandable_set** (Ewl_Model *m, Ewl_Model_Data_Expandable exp)
Sets the expandable callback on the given model.
- **Ewl_Model_Data_Fetch ewl_model_data_fetch_get** (Ewl_Model *m)
Gets the fetch callback from the model.
- **void ewl_model_data_fetch_set** (Ewl_Model *m, Ewl_Model_Data_Fetch get)
Sets the fetch callback in the model.
- **Ewl_Model_Data_Free ewl_model_data_free_get** (Ewl_Model *m)
Get the data free pointer set on this model.
- **void ewl_model_data_free_set** (Ewl_Model *m, Ewl_Model_Data_Free assign)
Set the data free pointer on this view.
- **Ewl_Model_Data_Header_Fetch ewl_model_data_header_fetch_get** (Ewl_Model *m)
Retrieves the header fetch function.

- **void ewl_model_data_header_fetch_set** (Ewl_Model *m, Ewl_Model_Data_Header_Fetch get)
Sets the fetch callback into the model.
- **Ewl_Model_Data_Sort ewl_model_data_sort_get** (Ewl_Model *m)
Gets the sort callback from the model.
- **void ewl_model_data_sort_set** (Ewl_Model *m, Ewl_Model_Data_Sort sort)
Sets the sort callback into the model.
- **Ewl_Model * ewl_model_ecore_list_get** (void)
Retrieves a model pre-initialized to work with an ecore list. This will setup the fetch and count methods for you.
- **Ewl_Model_Expansion_Data_Fetch ewl_model_expansion_data_fetch_get** (Ewl_Model *m)
Gets the expansion_data callback from the model.
- **void ewl_model_expansion_data_fetch_set** (Ewl_Model *m, Ewl_Model_Expansion_Data_Fetch get)
Sets the subfetch callback into the model.
- **Ewl_Model_Expansion_Model_Fetch ewl_model_expansion_model_fetch_get** (Ewl_Model *m)
Retrieves the model expansion model fetch function.
- **void ewl_model_expansion_model_fetch_set** (Ewl_Model *m, Ewl_Model_Expansion_Model_Fetch f)
Sets the model expansion model fetch callback to f.
- **int ewl_model_init** (Ewl_Model *model)
Set the model to the default values.
- **Ewl_Model * ewl_model_new** (void)
*Creates a new **Ewl_Model** (p. 703) object.*

6.48.2 Define Documentation

6.48.2.1 #define EWL_MODEL(model) ((Ewl_Model *)model)

Typecasts a pointer to an **Ewl_Model** (p. 703) pointer.

6.48.2.2 #define EWL_MODEL_COLUMN_SORTABLE(f) ((Ewl_Model_Column_Sortable)f)

Model callback to check if a columns data is sortable

6.48.2.3 `#define EWL_MODEL_DATA_COUNT(f) ((Ewl_Model_Data_Count)f)`

Model callback to have the program return the number of rows in its data

6.48.2.4 `#define EWL_MODEL_DATA_EXPANDABLE(f) ((Ewl_Model_Data_Expandable)f)`

Model callback to handle fetching the expandable flag for a given row of the tree

6.48.2.5 `#define EWL_MODEL_DATA_FETCH(f) ((Ewl_Model_Data_Fetch)f)`

Model callback to handle fetching the data at the given row/column

6.48.2.6 `#define EWL_MODEL_DATA_HEADER_FETCH(f) ((Ewl_Model_Data_Header_Fetch)f)`

Model callback to get the header data for a specific column

6.48.2.7 `#define EWL_MODEL_DATA_SORT(f) ((Ewl_Model_Data_Sort)f)`

Model callback to inform the program to sort it's data in the given column

6.48.2.8 `#define EWL_MODEL_EXPANSION_DATA_FETCH(f) ((Ewl_Model_Expansion_Data_Fetch)f)`

Model callback to get the expansion data for a specific row

6.48.2.9 `#define EWL_MODEL_EXPANSION_MODEL_FETCH(f) ((Ewl_Model_Expansion_Model_Fetch)f)`

Model callback to get the model to use for the expansion point

6.48.2.10 `#define EWL_MODEL_FREE(f) ((Ewl_Model_Data_Free)f)`

Model callback to free data after passing it to be assigned in the view.

6.48.3 Typedef Documentation

6.48.3.1 `typedef struct Ewl_Model Ewl_Model`

The `Ewl_Model` (p. 703) structure

6.48.3.2 `typedef int(*) Ewl_Model_Column_Sortable(void *data, int col)`

A typedef to shorten the definition of the `model_column_sortable` callbacks.

6.48.3.3 `typedef int(*) Ewl_Model_Data_Count(void *data)`

A typedef to shorten the definition of the `model_count` callbacks.

6.48.3.4 `typedef int(*) Ewl_Model_Data_Expandable(void *data, unsigned int row)`

A typedef to shorten the definition of the `model_expandable` callback

6.48.3.5 `typedef void*(*) Ewl_Model_Data_Fetch(void *data, unsigned int row, unsigned int column)`

A typedef to shorten the definition of the `model_fetch` callbacks.

6.48.3.6 `typedef void(*) Ewl_Model_Data_Free(void *data)`

A typedef to shorten the definition of the `model_data_free` callbacks.

6.48.3.7 `typedef void*(*) Ewl_Model_Data_Header_Fetch(void *data, unsigned int col)`

A typedef to shorten the definition of the `model_data_header_fetch` callback

6.48.3.8 `typedef void(*) Ewl_Model_Data_Sort(void *data, unsigned int column, Ewl_Sort_Direction sort)`

A typedef to shorten the definition of the `model_sort` callbacks.

6.48.3.9 `typedef void*(*) Ewl_Model_Expansion_Data_Fetch(void *data, unsigned int row)`

A typedef to shorten the definition of the `model_expansion_data` callback

6.48.3.10 `typedef Ewl_Model*(*) Ewl_Model_Expansion_Model_Fetch(void *data, unsigned int row)`

A typedef to shorten the definition of the `model_expansion_model_fetch` callback

6.48.4 **Function Documentation****6.48.4.1** `int ewl_model_cb_ecore_list_count (void * data)`

This will return the number of elements in the `ecore_list`.

Parameters:

data,: The `ecore_list` to get the count from

Returns:

Returns the number of elements in the list

6.48.4.2 `void* ewl_model_cb_ecore_list_fetch (void * data, unsigned int row, unsigned int col UNUSED)`

This will return the *row* data element from the list.

Parameters:

data,: The ecore_list to get the data from

row,: The row to get the data from

col,: UNUSED

Returns:

Returns the data at the given row

6.48.4.3 `Ewl_Model_Column_Sortable ewl_model_column_sortable_get (Ewl_Model * m)`

Retrieves the sortable callback on the model.

Parameters:

m,: the model to work with

Returns:

Returns the sortable callback on the model

6.48.4.4 `void ewl_model_column_sortable_set (Ewl_Model * m, Ewl_Model_Column_Sortable sortable)`

Sets the sortable callback.

Parameters:

m,: the model to work with

sortable,: the sortable callback to set

Returns:

Returns no value

6.48.4.5 `Ewl_Model_Data_Count ewl_model_data_count_get (Ewl_Model * m)`

Gets the count callback from the model.

Parameters:

m,: The **Ewl_Model** (p. 703) to get the count callback from

Returns:

Returns the **Ewl_Model_Count** callback set on the model or NULL if none set.

**6.48.4.6 void ewl_model_data_count_set (Ewl_Model * *m*,
Ewl_Model_Data_Count *count*)**

Sets the count callback into the model.

Parameters:

m,: The **Ewl_Model** (p. 703) to set the count callback on

count,: The count callback to set on the model

Returns:

Returns no value.

**6.48.4.7 Ewl_Model_Data_Expandable ewl_model_data_expandable_get
(Ewl_Model * *m*)**

Retrieves the expandable callback set on the model.

Parameters:

m,: The model to get the expandable callback from

Returns:

Returns the expandable callback set on the model

**6.48.4.8 void ewl_model_data_expandable_set (Ewl_Model * *m*,
Ewl_Model_Data_Expandable *exp*)**

Sets the expandable callback on the given model.

Parameters:

m,: The model to set the callback on

exp,: The expandable callback

Returns:

Returns no value

6.48.4.9 Ewl_Model_Data_Fetch `ewl_model_data_fetch_get (Ewl_Model * m)`

Gets the fetch callback from the model.

Parameters:

m,: The **Ewl_Model** (p. 703) to get the fetch callback from

Returns:

Returns the Ewl_Model_Fetch callback or NULL if none set

6.48.4.10 void ewl_model_data_fetch_set (**Ewl_Model** * *m*,
Ewl_Model_Data_Fetch *get*)

Sets the fetch callback in the model.

Parameters:

m,: The **Ewl_Model** (p. 703) to set the fetch callback into

get,: The Ewl_Model_Fetch callback to set

Returns:

Returns no value.

6.48.4.11 Ewl_Model_Data_Free `ewl_model_data_free_get (Ewl_Model * m)`

Get the data free pointer set on this model.

Parameters:

m,: The **Ewl_Model** (p. 703) to get the data free function from

Returns:

Returns the Ewl_Model_Data_Free set into the **Ewl_Model** (p. 703) or NULL if none set.

6.48.4.12 void ewl_model_data_free_set (**Ewl_Model** * *m*,
Ewl_Model_Data_Free *data_free*)

Set the data free pointer on this view.

Parameters:

m,: The **Ewl_Model** (p. 703) to set the data free function into

data_free,: The Ewl_Model_Data_Free function to set

Returns:

Returns no value.

This function is necessary to decrement intermediate reference counts or free temporary data buffers passed between the allocated data and the assigned widget.

6.48.4.13 Ewl_Model_Data_Header_Fetch `ewl_model_data_header_fetch_get (Ewl_Model * m)`

Retrieves the header fetch function.

Parameters:

m,: The model to work with

Returns:

Returns the header fetch function

6.48.4.14 void ewl_model_data_header_fetch_set `(Ewl_Model * m, Ewl_Model_Data_Header_Fetch get)`

Sets the fetch callback into the model.

Parameters:

m,: The model to work with

get,: The fetch callback to set

Returns:

Returns no value

6.48.4.15 Ewl_Model_Data_Sort `ewl_model_data_sort_get (Ewl_Model * m)`

Gets the sort callback from the model.

Parameters:

m,: The **Ewl_Model** (p. 703) to get the sort callback from

Returns:

Returns the **Ewl_Model_Sort** set on the model or NULL if none set

6.48.4.16 void ewl_model_data_sort_set `(Ewl_Model * m, Ewl_Model_Data_Sort sort)`

Sets the sort callback into the model.

Parameters:

m,: The **Ewl_Model** (p. 703) to set the sort callback on

sort,: The sort callback to set

Returns:

Returns no value.

6.48.4.17 Ewl_Model* ewl_model_ecore_list_get (void)

Retrieves a model pre-initialized to work with an ecore list. This will setup the fetch and count methods for you.

Returns:

Returns a model that is setup to work with an ecore_list

6.48.4.18 Ewl_Model_Expansion_Data_Fetch ewl_model_expansion_data_fetch_get (Ewl_Model * m)

Gets the expansion_data callback from the model.

Parameters:

m,: The **Ewl_Model** (p. 703) to get the expansion_data callback from

Returns:

Returns the Ewl_Model_Expansion_Data_Fetch for the model

6.48.4.19 void ewl_model_expansion_data_fetch_set (Ewl_Model * m, Ewl_Model_Expansion_Data_Fetch get)

Sets the subfetch callback into the model.

Parameters:

m,: The **Ewl_Model** (p. 703) to set the expansion_data callback into

get,: The Ewl_Model_Expansion_Data_Fetch callback for subrows

Returns:

Returns no value.

6.48.4.20 Ewl_Model_Expansion_Model_Fetch ewl_model_expansion_model_fetch_get (Ewl_Model * m)

Retrieves the model expansion model fetch function.

Parameters:

m,: The model to work with

Returns:

Returns the model expansion model fetch function

6.48.4.21 void ewl_model_expansion_model_fetch_set (Ewl_Model * *m*,
Ewl_Model_Expansion_Model_Fetch *f*)

Sets the model expansion model fetch callback to *f*.

Parameters:

m,: The model to work with

f,: The model expansion model fetch callback

Returns:

Returns no value

6.48.4.22 int ewl_model_init (Ewl_Model * *model*)

Set the model to the default values.

Parameters:

model,: The **Ewl_Model** (p. 703) to initialize

Returns:

Returns TRUE on success or FALSE on failure

6.48.4.23 Ewl_Model* ewl_model_new (void)

Creates a new **Ewl_Model** (p. 703) object.

Returns:

Returns a new **Ewl_Model** (p. 703) on success or NULL on failure

6.49 Ewl_MVC: The base model-view-controller framework

6.49.1 Detailed Description

Defines a widget to base MVC widgets off of

Remarks:

Inherits from **Ewl_Box** (p. 586).

Data Structures

- struct **Ewl_MVC**
*Inherits from **Ewl_Box** (p. 586) and extends to provide mvc functionality.*
- struct **Ewl_Selection**
Structure to store information on selections.
- struct **Ewl_Selection_Idx**
Structure to store information on a single index selection.
- struct **Ewl_Selection_Range**
Structure to store a range of selected cells.

Defines

- `#define EWL_MVC(mvc) ((Ewl_MVC *)mvc)`
- `#define EWL_MVC_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_MVC_TYPE))`
- `#define EWL_MVC_TYPE "mvc"`
- `#define EWL_SELECTION(x) ((Ewl_Selection *)x)`
- `#define EWL_SELECTION_IDX(x) ((Ewl_Selection_Idx *)x)`
- `#define EWL_SELECTION_RANGE(x) ((Ewl_Selection_Range *)x)`

Typedefs

- `typedef Ewl_MVC Ewl_MVC`
- `typedef Ewl_Selection Ewl_Selection`
- `typedef Ewl_Selection_Idx Ewl_Selection_Idx`
- `typedef Ewl_Selection_Range Ewl_Selection_Range`

Functions

- `void ewl_mvc_cb_destroy (Ewl_Widget *w, void *ev, void *data)`
- `void * ewl_mvc_data_get (Ewl_MVC *mvc)`
Retrieves the data set into the MVC widget.
- `void ewl_mvc_data_set (Ewl_MVC *mvc, void *data)`

Sets the given data data into the MVC widget mvc.

- unsigned int **ewl_mvc_dirty_get** (Ewl_MVC *mvc)
Retrieves the dirty status of the MVC widget.
- void **ewl_mvc_dirty_set** (Ewl_MVC *mvc, unsigned int dirty)
Sets the dirty status of the MVC widget mvc to the dirty state.
- void **ewl_mvc_handle_click** (Ewl_MVC *mvc, Ewl_Model *model, void *data, int row, int column)
- void **ewl_mvc_highlight** (Ewl_MVC *mvc, Ewl_Container *c, Ewl_Widget *(*widget)(Ewl_MVC *mvc, void *data, int row, int column))
- int **ewl_mvc_init** (Ewl_MVC *mvc)
Initializes an MVC widget ot default values.
- Ewl_Model * **ewl_mvc_model_get** (Ewl_MVC *mvc)
Retrieves the model set into the MVC widget.
- void **ewl_mvc_model_set** (Ewl_MVC *mvc, Ewl_Model *model)
Sets the given model into the tree.
- void **ewl_mvc_selected_add** (Ewl_MVC *mvc, Ewl_Model *model, void *data, int row, int column)
Adds the given index to the selected list.
- void **ewl_mvc_selected_change_cb_set** (Ewl_MVC *mvc, void(*cb)(Ewl_MVC *mvc))
- void **ewl_mvc_selected_clear** (Ewl_MVC *mvc)
clears the selection list
- int **ewl_mvc_selected_count_get** (Ewl_MVC *mvc)
Retrives the number of items selected in the widget.
- Ewl_Selection_Idx * **ewl_mvc_selected_get** (Ewl_MVC *mvc)
Retrieves the last selected item. Return must be free'd.
- unsigned int **ewl_mvc_selected_is** (Ewl_MVC *mvc, void *data, int row, int column)
- Ecore_List * **ewl_mvc_selected_list_get** (Ewl_MVC *mvc)
Retrieves the list of selected indicies. DO NOT remove or change items in this list.
- void **ewl_mvc_selected_list_set** (Ewl_MVC *mvc, Ecore_List *list)
Sets the list of items to select. This will remove any items it needs from the list.
- void **ewl_mvc_selected_range_add** (Ewl_MVC *mvc, Ewl_Model *model, void *data, int srow, int scolumn, int erow, int ecolumn)
Sets the given range, inclusive, as selected in the mvc.
- void **ewl_mvc_selected_rm** (Ewl_MVC *mvc, void *data, int row, int column)
- void **ewl_mvc_selected_set** (Ewl_MVC *mvc, Ewl_Model *model, void *data, int row, int column)

Sets the given index as selected.

- **Ewl_Selection *ewl_mvc_selection_index_new** (**Ewl_Model** *model, void *data, int row, int column)

Creates a new index selection based on given values.

- **Ewl_Selection_Mode ewl_mvc_selection_mode_get** (**Ewl_MVC** *mvc)

Retrieves the selection mode of the widget.

- void **ewl_mvc_selection_mode_set** (**Ewl_MVC** *mvc, **Ewl_Selection_Mode** mode)

Sets the selection capabilities of the mvc widget.

- **Ewl_Selection *ewl_mvc_selection_range_new** (**Ewl_Model** *model, void *data, int srow, int scolumn, int erow, int ecolum)

Creates a new range selection based on given values.

- void **ewl_mvc_view_change_cb_set** (**Ewl_MVC** *mvc, void(*cb)(**Ewl_MVC** *mvc))

This callback will be called whenever the ewl_mvc_view_set routine is called to notify the inheriting widget that the view has changed.

- **Ewl_View *ewl_mvc_view_get** (**Ewl_MVC** *mvc)

Retrives the current view set on the MVC.

- void **ewl_mvc_view_set** (**Ewl_MVC** *mvc, **Ewl_View** *view)

Sets the given view onto the MVC.

6.49.2 Define Documentation

6.49.2.1 #define EWL_MVC(mvc) ((Ewl_MVC *)mvc)

Typecasts a pointer to an **Ewl_MVC** (p. 705) pointer.

6.49.2.2 #define EWL_MVC_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_MVC_TYPE))

Returns TRUE if the widget is an **Ewl_Mvc**, FALSE otherwise

6.49.2.3 #define EWL_MVC_TYPE "mvc"

The type name for the **Ewl_MVC** (p. 705) widget

6.49.2.4 #define EWL_SELECTION(x) ((Ewl_Selection *)x)

Typecast a pointer to an **Ewl_Selection** (p. 736) pointer

6.49.2.5 `#define EWL_SELECTION_IDX(x) ((Ewl_Selection_Idx *)x)`

Typecast a pointer to an `Ewl_Selection_Idx` (p. 737) pointer

6.49.2.6 `#define EWL_SELECTION_RANGE(x) ((Ewl_Selection_Range *)x)`

Typecasts a pointer to an `Ewl_Selection_Range` (p. 738)

6.49.3 Typedef Documentation**6.49.3.1** `typedef struct Ewl_MVC Ewl_MVC`

A simple mvc base class

6.49.3.2 `typedef struct Ewl_Selection Ewl_Selection`

`Ewl_Selection` (p. 736)

6.49.3.3 `typedef struct Ewl_Selection_Idx Ewl_Selection_Idx`

`Ewl_Selection_Idx` (p. 737)

6.49.3.4 `typedef struct Ewl_Selection_Range Ewl_Selection_Range`

`Ewl_Selection_Range` (p. 738)

6.49.4 Function Documentation**6.49.4.1** `void ewl_mvc_cb_destroy (Ewl_Widget * w, void * ev, void * data)`**6.49.4.2** `void* ewl_mvc_data_get (Ewl_MVC * mvc)`

Retrieves the data set into the MVC widget.

Parameters:

mvc,: The MVC to work with

Returns:

Returns the data set onto the MVC widget

6.49.4.3 `void ewl_mvc_data_set (Ewl_MVC * mvc, void * data)`

Sets the given data *data* into the MVC widget *mvc*.

Parameters:

mvc,: The MVC to work with

data,: The data to set on the MVC

Returns:

Returns no value

6.49.4.4 unsigned int ewl_mvc_dirty_get (Ewl_MVC * *mvc*)

Retrieves the dirty status of the MVC widget.

Parameters:

mvc,: The MVC widget use

Returns:

Returns the dirty status of the MVC widget

6.49.4.5 void ewl_mvc_dirty_set (Ewl_MVC * *mvc*, unsigned int *dirty*)

Sets the dirty status of the MVC widget *mvc* to the *dirty* state.

Parameters:

mvc,: The MVC widget to work with

dirty,: The dirty status to set

Returns:

Returns no value.

6.49.4.6 void ewl_mvc_handle_click (Ewl_MVC * *mvc*, Ewl_Model * *model*, void * *data*, int *row*, int *column*)

6.49.4.7 void ewl_mvc_highlight (Ewl_MVC * *mvc*, Ewl_Container * *c*, Ewl_Widget *(*)(Ewl_MVC **mvc*, void **data*, int *row*, int *column*) *widget*)

6.49.4.8 int ewl_mvc_init (Ewl_MVC * *mvc*)

Initializes an MVC widget ot default values.

Parameters:

mvc,: The MVC to initialize

Returns:

Returns TRUE on success or FALSE if unsuccessful

6.49.4.9 Ewl_Model* ewl_mvc_model_get (Ewl_MVC * *mvc*)

Retrieves the model set into the MVC widget.

Parameters:

mvc,: The MVC to work with

Returns:

Returns the current model set into the MVC widget

6.49.4.10 void ewl_mvc_model_set (Ewl_MVC * *mvc*, Ewl_Model * *model*)

Sets the given model into the tree.

Parameters:

mvc,: The MVC to work with

model,: The model to set

Returns:

Returns no value

6.49.4.11 void ewl_mvc_selected_add (Ewl_MVC * *mvc*, Ewl_Model * *model*, void * *data*, int *row*, int *column*)

Adds the given index to the selected list.

Parameters:

mvc,: The MVC to work with

model,: The model to work with. If NULL the model from the MVC will be used

data,: The parent data containing the index selection

row,: The row to add

column,: The column to add

Returns:

Returns no value

6.49.4.12 void ewl_mvc_selected_change_cb_set (Ewl_MVC * *mvc*, void(*) (Ewl_MVC **mvc*) *cb*)**6.49.4.13 void ewl_mvc_selected_clear (Ewl_MVC * *mvc*)**

clears the selection list

Parameters:

mvc,: The mvc to clear

Returns:

Returns no value

6.49.4.14 `int ewl_mvc_selected_count_get (Ewl_MVC * mvc)`

Retrives the number of items selected in the widget.

Parameters:

mvc,: The MVC widget to work with

Returns:

Returns the number of items selected in the MVC

6.49.4.15 `Ewl_Selection_Idx* ewl_mvc_selected_get (Ewl_MVC * mvc)`

Retrieves the last selected item. Return must be free'd.

Parameters:

mvc,: The MVC to get the data from

Returns:

Returns the last selected item. Return must be free'd

6.49.4.16 `unsigned int ewl_mvc_selected_is (Ewl_MVC * mvc, void * data, int row, int column)`**6.49.4.17** `Ecore_List* ewl_mvc_selected_list_get (Ewl_MVC * mvc)`

Retrieves the list of selected indicies. DO NOT remove or change items in this list.

Parameters:

mvc,: The MVC to get the list from

Returns:

Returns the list of selected indices

6.49.4.18 `void ewl_mvc_selected_list_set (Ewl_MVC * mvc, Ecore_List * list)`

Sets the list of items to select. This will remove any items it needs from the list.

Parameters:

mvc,: The MVC to work with

list,: The list of items to set selected.

Returns:

Returns no value

6.49.4.19 void ewl_mvc_selected_range_add (Ewl_MVC * *mvc*, Ewl_Model * *model*, void * *data*, int *srow*, int *scolumn*, int *erow*, int *ecolumn*)

Sets the given range, inclusive, as selected in the mvc.

Parameters:

mvc,: The MVC to set the list into

model,: The model to use for this data. If NULL the model from the MVC will be used

data,: The parent data containing the index selection

srow,: The start row

scolumn,: The start column

erow,: The end row

ecolumn,: The end column

Returns:

Returns no value

6.49.4.20 void ewl_mvc_selected_rm (Ewl_MVC * *mvc*, void * *data*, int *row*, int *column*)

6.49.4.21 void ewl_mvc_selected_set (Ewl_MVC * *mvc*, Ewl_Model * *model*, void * *data*, int *row*, int *column*)

Sets the given index as selected.

Parameters:

mvc,: The MVC to work with

model,: The model to work with the data. If NULL the model from the MVC will be used

data,: The parent data containing the index selection

row,: The row to set

column,: The column to set

Returns:

Returns no value

6.49.4.22 Ewl_Selection* ewl_mvc_selection_index_new (Ewl_Model * *model*, void * *data*, int *row*, int *column*)

Creates a new index selection based on given values.

Parameters:

model,: The model to work with this data

data,: The parent data containing the index selection

row,: The row to create the index selection for

column,: The column to create the index for

Returns:

Returns a new **Ewl_Selection_Idx** (p. 737) based on the *row* and *column*

6.49.4.23 Ewl_Selection_Mode ewl_mvc_selection_mode_get (Ewl_MVC * *mvc*)

Retrieves the selection mode of the widget.

Parameters:

mvc,: The MVC widget to use

Returns:

Returns the selection mode of the mvc widget

6.49.4.24 void ewl_mvc_selection_mode_set (Ewl_MVC * *mvc*, Ewl_Selection_Mode *mode*)

Sets the selection capabilities of the mvc widget.

Parameters:

mvc,: The MVC widget to use

mode,: The selection mode to set

Returns:

Returns no value

6.49.4.25 Ewl_Selection* ewl_mvc_selection_range_new (Ewl_Model * *model*, void * *data*, int *srow*, int *scolumn*, int *erow*, int *ecolumn*)

Creates a new range selection based on given values.

Parameters:

model,: The model to work with this data

data,: The data that we're working with

srow,: The start row

scolumn,: The start column

erow,: The end row

ecolumn,: The end column

Returns:

Returns a new **Ewl_Selection_Range** (p. 738) based on given values

6.49.4.26 `void ewl_mvc_view_change_cb_set (Ewl_MVC * mvc,
void(*) (Ewl_MVC *mvc) cb)`

This callback will be called whenever the `ewl_mvc_view_set` routine is called to notify the inheriting widget that the view has changed.

Parameters:

mvc,: The MVC to work with

cb,: The callback to set

Returns:

Returns no value

6.49.4.27 `Ewl_View* ewl_mvc_view_get (Ewl_MVC * mvc)`

Retrives the current view set on the MVC.

Parameters:

mvc,: The MVC to work with

Returns:

Returns the current view set on the MVC

6.49.4.28 `void ewl_mvc_view_set (Ewl_MVC * mvc, Ewl_View * view)`

Sets the given view onto the MVC.

Parameters:

mvc,: The MVC to work with

view,: The view to set

Returns:

Returns no value

6.50 Ewl_Notebook: A notebook widget

6.50.1 Detailed Description

Defines a class for notebook layout

Remarks:

Inherits from **Ewl_Box** (p. 586).

Data Structures

- struct **Ewl_Notebook**

*Inherits from **Ewl_Box** (p. 586) and extends to provide for a notebook style layout.*

Defines

- #define **EWL_NOTEBOOK**(n) ((**Ewl_Notebook** *)n)
- #define **EWL_NOTEBOOK_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_NOTEBOOK_TYPE**))
- #define **EWL_NOTEBOOK_TYPE** "notebook"

Typedefs

- typedef **Ewl_Notebook** **Ewl_Notebook**

Functions

- void **ewl_notebook_cb_child_add** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_notebook_cb_child_hide** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_notebook_cb_child_remove** (**Ewl_Container** *c, **Ewl_Widget** *w, int rem_idx)
- void **ewl_notebook_cb_child_show** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_notebook_cb_tab_clicked** (**Ewl_Widget** *w, void *ev, void *data)
- int **ewl_notebook_init** (**Ewl_Notebook** *n)

Initialize a notebook to default values and callbacks.

- **Ewl_Widget** * **ewl_notebook_new** (void)

Create a new notebook widget.

- const char * **ewl_notebook_page_tab_text_get** (**Ewl_Notebook** *n, **Ewl_Widget** *page)

Get the text of the notebook page page.

- void **ewl_notebook_page_tab_text_set** (**Ewl_Notebook** *n, **Ewl_Widget** *page, const char *text)

Set the text of the tab for the page page to the text text.

- **Ewl_Widget * ewl_notebook_page_tab_widget_get** (**Ewl_Notebook *n**, **Ewl_Widget *page**)
*Retrieve the widget used as the tab for the page **page**.*
- **void ewl_notebook_page_tab_widget_set** (**Ewl_Notebook *n**, **Ewl_Widget *page**, **Ewl_Widget *tab**)
*Set the widget to use as the tab for the page **page** to widget **tab**.*
- **unsigned int ewl_notebook_tabbar_alignment_get** (**Ewl_Notebook *n**)
Retrieves the alignment of the tabbar in the notebook widget.
- **void ewl_notebook_tabbar_alignment_set** (**Ewl_Notebook *n**, **unsigned int align**)
Set the alignment of the tabbar in the notebook widget.
- **Ewl_Position ewl_notebook_tabbar_position_get** (**Ewl_Notebook *n**)
Get the position of the tabbar in the notebook widget.
- **void ewl_notebook_tabbar_position_set** (**Ewl_Notebook *n**, **Ewl_Position pos**)
Set the position of the tabbar in the notebook widget.
- **unsigned int ewl_notebook_tabbar_visible_get** (**Ewl_Notebook *n**)
Retrieve if the tabbar is visible.
- **void ewl_notebook_tabbar_visible_set** (**Ewl_Notebook *n**, **unsigned int visible**)
Set if the tabbar is visible.
- **Ewl_Widget * ewl_notebook_visible_page_get** (**Ewl_Notebook *n**)
Get the current visible page of the notebook.
- **void ewl_notebook_visible_page_set** (**Ewl_Notebook *n**, **Ewl_Widget *page**)
Set the current visible page of the notebook.

6.50.2 Define Documentation

6.50.2.1 #define EWL_NOTEBOOK(n) ((Ewl_Notebook *)n)

Typecasts a pointer to an **Ewl_Notebook** (p. 707) pointer.

6.50.2.2 #define EWL_NOTEBOOK_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_NOTEBOOK_TYPE))

Returns TRUE if the widget is an **Ewl_Notebook** (p. 707), FALSE otherwise

6.50.2.3 #define EWL_NOTEBOOK_TYPE "notebook"

The type name for the **Ewl_Notebook** (p. 707) widget

6.50.3 Typedef Documentation

6.50.3.1 typedef struct Ewl_Notebook Ewl_Notebook

Holds the notebook information

6.50.4 Function Documentation

6.50.4.1 void ewl_notebook_cb_child_add (Ewl_Container * *c*, Ewl_Widget * *w*)

6.50.4.2 void ewl_notebook_cb_child_hide (Ewl_Container * *c*, Ewl_Widget * *w*)

6.50.4.3 void ewl_notebook_cb_child_remove (Ewl_Container * *c*, Ewl_Widget * *w*, int *rem_idx*)

6.50.4.4 void ewl_notebook_cb_child_show (Ewl_Container * *c*, Ewl_Widget * *w*)

6.50.4.5 void ewl_notebook_cb_tab_clicked (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.50.4.6 int ewl_notebook_init (Ewl_Notebook * *n*)

Initialize a notebook to default values and callbacks.

Parameters:

n,: The **Ewl_Notebook** (p. 707) widget to initialize

Returns:

Returns TRUE on success or NULL on failure.

6.50.4.7 Ewl_Widget* ewl_notebook_new (void)

Create a new notebook widget.

Returns:

Returns a newly allocated notebook on success. NULL on failure

6.50.4.8 const char* ewl_notebook_page_tab_text_get (Ewl_Notebook * *n*, Ewl_Widget * *page*)

Get the text of the notebook page *page*.

Parameters:

n,: The **Ewl_Notebook** (p. 707) to work with

page,: The page to get the tab text from

Returns:

Returns the text of the pages tab

6.50.4.9 void ewl_notebook_page_tab_text_set (Ewl_Notebook * *n*,
Ewl_Widget * *page*, const char * *text*)

Set the text of the tab for the page *page* to the text *text*.

Parameters:

n,: The **Ewl_Notebook** (p. 707) to set the tab text in

page,: The page to associate the tab text too

text,: The text to set in the tab

Returns:

Returns no value.

6.50.4.10 Ewl_Widget* ewl_notebook_page_tab_widget_get (Ewl_Notebook *
n, Ewl_Widget * *page*)

Retrieve the widget used as the tab for the page *page*.

Parameters:

n,: The **Ewl_Notebook** (p. 707) to get the tab widget from

page,: The page to get the tab from

Returns:

Returns the tab widget associated with the given page

6.50.4.11 void ewl_notebook_page_tab_widget_set (Ewl_Notebook * *n*,
Ewl_Widget * *page*, Ewl_Widget * *tab*)

Set the widget to use as the tab for the page *page* to widget *tab*.

Parameters:

n,: The **Ewl_Notebook** (p. 707) to set the tab into

page,: The page to associate the tab with

tab,: The contents of the tab

Returns:

Returns no value.

6.50.4.12 `unsigned int ewl_notebook_tabbar_alignment_get (Ewl_Notebook * n)`

Retrieves the alignment of the tabbar in the notebook widget.

Parameters:

n,: The **Ewl_Notebook** (p. 707) to get the alignment from

Returns:

Returns the current alignment values of the widget

6.50.4.13 `void ewl_notebook_tabbar_alignment_set (Ewl_Notebook * n, unsigned int align)`

Set the alignment of the tabbar in the notebook widget.

Parameters:

n,: The **Ewl_Notebook** (p. 707) to set the tabbar alignment of

align,: The **Ewl_Alignment** to set the alignment too

Returns:

Returns no value.

6.50.4.14 `Ewl_Position ewl_notebook_tabbar_position_get (Ewl_Notebook * n)`

Get the position of the tabbar in the notebook widget.

Parameters:

n,: The **Ewl_Notebook** (p. 707) to get the tabbar position of

Returns:

Returns the current **Ewl_Position** of the tabbar

6.50.4.15 `void ewl_notebook_tabbar_position_set (Ewl_Notebook * n, Ewl_Position pos)`

Set the position of the tabbar in the notebook widget.

Parameters:

n,: The **Ewl_Notebook** (p. 707) to set the tabbar position of

pos,: The **Ewl_Position** to put the tabbar in

Returns:

Returns no value.

6.50.4.16 unsigned int ewl_notebook_tabbar_visible_get (Ewl_Notebook * *n*)

Retrieve if the tabbar is visible.

Parameters:

n,: The **Ewl_Notebook** (p. 707) to get the tabbar visibility of

Returns:

Returns the visibility of the tabbar

6.50.4.17 void ewl_notebook_tabbar_visible_set (Ewl_Notebook * *n*, unsigned int *visible*)

Set if the tabbar is visible.

Parameters:

n,: The **Ewl_Notebook** (p. 707) to set the visible flag of the tabbar

visible,: The flag to set for the tabbar visibility

Returns:

Returns no value.

6.50.4.18 Ewl_Widget* ewl_notebook_visible_page_get (Ewl_Notebook * *n*)

Get the current visible page of the notebook.

Parameters:

n,: The **Ewl_Notebook** (p. 707) to get the visible page of

Returns:

Returns the visible page of the notebook

6.50.4.19 void ewl_notebook_visible_page_set (Ewl_Notebook * *n*, Ewl_Widget * *page*)

Set the current visible page of the notebook.

Parameters:

n,: The **Ewl_Notebook** (p. 707) to set the visible page of

page,: The page to set visible in the tabbar

Returns:

Returns no value.

6.51 Ewl_Object: Base object. Provides functionality available to all widgets.

6.51.1 Detailed Description

Defines the **Ewl_Object** (p. 709) class along with basic methods and macros.

Data Structures

- struct **Ewl_Object**

Provides facilities for sizing, position, alignment and fill policy.

Defines

- #define **CONFIGURED**(o)
- #define **CURRENT_H**(o) EWL_OBJECT(o) → current.h
- #define **CURRENT_W**(o) EWL_OBJECT(o) → current.w
- #define **CURRENT_X**(o) EWL_OBJECT(o) → current.x
- #define **CURRENT_Y**(o) EWL_OBJECT(o) → current.y
- #define **DESTROYED**(o)
- #define **DISABLED**(o) (ewl_object_state_has(EWL_OBJECT(o), EWL_FLAG_STATE_DISABLED))
- #define **EWL_OBJECT**(object) ((**Ewl_Object** *) object)
- #define **ewl_object_custom_h_set**(o, h)
- #define **ewl_object_custom_size_set**(o, w, h)
- #define **ewl_object_custom_w_set**(o, w)
- #define **ewl_object_flags_get**(o, mask) (EWL_OBJECT(o) → flags & mask)
Retrieves the current setting of the object flags for o.
- #define **ewl_object_flags_has**(o, check_flags, mask) (!! (EWL_OBJECT(o) → flags & ((check_flags) & mask)))
Determines if an object has the requested flags set.
- #define **ewl_object_flags_has_all**(o, check_flags, mask) ((EWL_OBJECT(o) → flags & ((check_flags) & mask)) == ((check_flags) & mask))
Determines if an object has all of the requested flags set.
- #define **ewl_object_in_tab_list_get**(o) (ewl_object_flags_get(o, EWL_FLAG_PROPERTY_IN_TAB_LIST))
Retrieves the current setting of the in tab list flag for o.
- #define **ewl_object_in_tab_list_set**(o, val)
Changes the tab list flag value to match val.
- #define **EWL_OBJECT_MAX_SIZE** (INT_MAX)
- #define **EWL_OBJECT_MIN_SIZE** (1)
- #define **ewl_object_queued_add**(o, queued) ewl_object_flags_add(o, queued, EWL_FLAGS_QUEUED_MASK)

- `#define ewl_object_queued_get(o, queued) ewl_object_flags_get(o, queued, EWL_FLAGS_QUEUED_MASK)`
- `#define ewl_object_queued_has(o, queued) ewl_object_flags_has(o, queued, EWL_FLAGS_QUEUED_MASK)`
- `#define ewl_object_queued_remove(o, queued) ewl_object_flags_remove(o, queued, EWL_FLAGS_QUEUED_MASK)`
- `#define ewl_object_recursive_get(o) (ewl_object_flags_get(o, EWL_FLAG_PROPERTY_RECURSIVE))`

Retrieves the current setting of the recursive flag for o.

- `#define ewl_object_recursive_set(o, val)`

Changes the recursive flag value to match val.

- `#define ewl_object_state_add(o, state) ewl_object_flags_add(o, state, EWL_FLAGS_STATE_MASK)`
- `#define ewl_object_state_get(o, state) ewl_object_flags_get(o, state, EWL_FLAGS_STATE_MASK)`
- `#define ewl_object_state_has(o, state) ewl_object_flags_has(o, state, EWL_FLAGS_STATE_MASK)`
- `#define ewl_object_state_remove(o, state) ewl_object_flags_remove(o, state, EWL_FLAGS_STATE_MASK)`
- `#define ewl_object_toplevel_get(o) (ewl_object_flags_get(o, EWL_FLAG_PROPERTY_TOPLEVEL))`

Retrieves the current setting of the top level flag for o.

- `#define ewl_object_toplevel_set(o, val)`

Changes the top level flag value to match val.

- `#define ewl_object_visible_add(o, visible) ewl_object_flags_add(o, visible, EWL_FLAGS_VISIBLE_MASK)`
- `#define ewl_object_visible_get(o, visible) ewl_object_flags_get(o, visible, EWL_FLAGS_VISIBLE_MASK)`
- `#define ewl_object_visible_has(o, visible) ewl_object_flags_has(o, visible, EWL_FLAGS_VISIBLE_MASK)`
- `#define ewl_object_visible_remove(o, visible) ewl_object_flags_remove(o, visible, EWL_FLAGS_VISIBLE_MASK)`
- `#define HIDDEN(o) (!(EWL_OBJECT(o) → flags & EWL_FLAG_VISIBLE_SHOWN))`
- `#define INSET_BOTTOM(o) EWL_OBJECT(o) → insets.b`
- `#define INSET_HORIZONTAL(o) (EWL_OBJECT(o) → insets.l + EWL_OBJECT(o) → insets.r)`
- `#define INSET_LEFT(o) EWL_OBJECT(o) → insets.l`
- `#define INSET_RIGHT(o) EWL_OBJECT(o) → insets.r`
- `#define INSET_TOP(o) EWL_OBJECT(o) → insets.t`
- `#define INSET_VERTICAL(o) (EWL_OBJECT(o) → insets.t + EWL_OBJECT(o) → insets.b)`
- `#define MAXIMUM_H(o) EWL_OBJECT(o) → maximum.h`
- `#define MAXIMUM_W(o) EWL_OBJECT(o) → maximum.w`
- `#define MINIMUM_H(o) EWL_OBJECT(o) → minimum.h`
- `#define MINIMUM_W(o) EWL_OBJECT(o) → minimum.w`

- `#define OBSCURED(o) (EWL_OBJECT(o) → flags & EWL_FLAG_VISIBLE_-OBSCURED)`
- `#define PADDING_BOTTOM(o) EWL_OBJECT(o) → pad.b`
- `#define PADDING_HORIZONTAL(o) (EWL_OBJECT(o) → pad.l + EWL_OBJECT(o) → pad.r)`
- `#define PADDING_LEFT(o) EWL_OBJECT(o) → pad.l`
- `#define PADDING_RIGHT(o) EWL_OBJECT(o) → pad.r`
- `#define PADDING_TOP(o) EWL_OBJECT(o) → pad.t`
- `#define PADDING_VERTICAL(o) (EWL_OBJECT(o) → pad.t + EWL_OBJECT(o) → pad.b)`
- `#define PREFERRED_H(o) EWL_OBJECT(o) → preferred.h`
- `#define PREFERRED_W(o) EWL_OBJECT(o) → preferred.w`
- `#define REALIZED(o) (EWL_OBJECT(o) → flags & EWL_FLAG_VISIBLE_-REALIZED)`
- `#define RECURSIVE(o) (EWL_OBJECT(o) → flags & EWL_FLAG_PROPERTY_-RECURSIVE)`
- `#define VISIBLE(o) (EWL_OBJECT(o) → flags & EWL_FLAG_VISIBLE_SHOWN)`

Typedefs

- `typedef Ewl_Object Ewl_Object`

Functions

- `unsigned int ewl_object_alignment_get (Ewl_Object *o)`
Retrieves the current setting of the object alignment for o.
- `void ewl_object_alignment_set (Ewl_Object *o, unsigned int align)`
Change the alignment of the specified object.
- `void ewl_object_current_geometry_get (Ewl_Object *o, int *x, int *y, int *w, int *h)`
Retrieve the size and position of object.
- `int ewl_object_current_h_get (Ewl_Object *o)`
Get the current height of the object.
- `void ewl_object_current_size_get (Ewl_Object *o, int *w, int *h)`
Retrieve the current size of an object.
- `int ewl_object_current_w_get (Ewl_Object *o)`
Get the current width of the object.
- `int ewl_object_current_x_get (Ewl_Object *o)`
get the current x position of the object
- `int ewl_object_current_y_get (Ewl_Object *o)`
Get the current y position of the object.
- `unsigned int ewl_object_fill_policy_get (Ewl_Object *o)`

Retrieves the current setting of the object fill policy for o.

- void **ewl_object_fill_policy_set** (Ewl_Object *o, unsigned int fill)
Change the fill policy of the specified object.
- void **ewl_object_flags_add** (Ewl_Object *o, unsigned int flags, unsigned int mask)
Add the set of flags specified in flags to o.
- void **ewl_object_flags_remove** (Ewl_Object *o, unsigned int flags, unsigned int mask)
Removes the set of state flags specified in flags from o.
- void **ewl_object_geometry_request** (Ewl_Object *o, int x, int y, int w, int h)
Request a new geometry be applied to an object.
- void **ewl_object_h_request** (Ewl_Object *o, int h)
Request a new width for an object.
- int **ewl_object_init** (Ewl_Object *o)
- int **ewl_object_insets_bottom_get** (Ewl_Object *o)
Get the bottom insets for the object.
- void **ewl_object_insets_get** (Ewl_Object *o, int *l, int *r, int *t, int *b)
Retrieve the edge insets of an object.
- int **ewl_object_insets_left_get** (Ewl_Object *o)
Get the left insets for the object.
- int **ewl_object_insets_right_get** (Ewl_Object *o)
Get the right insets for the object.
- void **ewl_object_insets_set** (Ewl_Object *o, int l, int r, int t, int b)
Set the insets around an objects edges.
- int **ewl_object_insets_top_get** (Ewl_Object *o)
Get the top insets for the object.
- int **ewl_object_maximum_h_get** (Ewl_Object *o)
Get the maximum height of an object.
- void **ewl_object_maximum_h_set** (Ewl_Object *o, int h)
Set the minimum height of an object.
- void **ewl_object_maximum_size_get** (Ewl_Object *o, int *w, int *h)
Retrieve the minimum dimensions of an object.
- void **ewl_object_maximum_size_set** (Ewl_Object *o, int w, int h)
Set the maximum size of an object.
- int **ewl_object_maximum_w_get** (Ewl_Object *o)

Get the maximum width of an object.

- `void ewl_object_maximum_w_set (Ewl_Object *o, int w)`
Set the minimum width of an object.
- `int ewl_object_minimum_h_get (Ewl_Object *o)`
Get the minimum height of an object.
- `void ewl_object_minimum_h_set (Ewl_Object *o, int h)`
Set the minimum height of an object.
- `void ewl_object_minimum_size_get (Ewl_Object *o, int *w, int *h)`
Retrieve the minimum dimensions of an object.
- `void ewl_object_minimum_size_set (Ewl_Object *o, int w, int h)`
Set the minimum size of an object.
- `int ewl_object_minimum_w_get (Ewl_Object *o)`
Get the minimum width of an object.
- `void ewl_object_minimum_w_set (Ewl_Object *o, int w)`
Set the minimum width of an object.
- `int ewl_object_padding_bottom_get (Ewl_Object *o)`
Get the bottom padding for the object.
- `void ewl_object_padding_get (Ewl_Object *o, int *l, int *r, int *t, int *b)`
Retrieve the edge padding of an object.
- `int ewl_object_padding_left_get (Ewl_Object *o)`
Get the left padding for the object.
- `int ewl_object_padding_right_get (Ewl_Object *o)`
Get the right padding for the object.
- `void ewl_object_padding_set (Ewl_Object *o, int l, int r, int t, int b)`
Set the padding around an objects edges.
- `int ewl_object_padding_top_get (Ewl_Object *o)`
Get the top padding for the object.
- `void ewl_object_place (Ewl_Object *o, int x, int y, int w, int h)`
Assign a specific area to an object.
- `void ewl_object_position_request (Ewl_Object *o, int x, int y)`
Request a new position be applied to an object.
- `int ewl_object_preferred_h_get (Ewl_Object *o)`
Get the preferred height of the object.

- `int ewl_object_preferred_inner_h_get (Ewl_Object *o)`
Get the preferred height of the object.
- `void ewl_object_preferred_inner_h_set (Ewl_Object *o, int h)`
Set the preferred height of the object.
- `void ewl_object_preferred_inner_size_get (Ewl_Object *o, int *w, int *h)`
Retrieve the preferred size of an object.
- `void ewl_object_preferred_inner_size_set (Ewl_Object *o, int w, int h)`
Set the preferred size of the object.
- `int ewl_object_preferred_inner_w_get (Ewl_Object *o)`
Get the preferred width of the object, ignoring padding and insets.
- `void ewl_object_preferred_inner_w_set (Ewl_Object *o, int w)`
Set the preferred width of the object.
- `void ewl_object_preferred_size_get (Ewl_Object *o, int *w, int *h)`
Retrieve the preferred size of an object.
- `int ewl_object_preferred_w_get (Ewl_Object *o)`
Get the preferred width of the object.
- `void ewl_object_size_request (Ewl_Object *o, int w, int h)`
Request a new size be applied to an object.
- `void ewl_object_w_request (Ewl_Object *o, int w)`
Request a new width for an object.
- `void ewl_object_x_request (Ewl_Object *o, int x)`
Request a new x position for an object.
- `void ewl_object_y_request (Ewl_Object *o, int y)`
Request a new y position for an object.

6.51.2 Define Documentation

6.51.2.1 #define CONFIGURED(o)

Value:

```
(ewl_object_queued_has(EWL_OBJECT(o), \
                        EWL_FLAG_QUEUED_CSCHEDULED) \
    || ewl_object_queued_has(EWL_OBJECT(o), \
                        EWL_FLAG_QUEUED_CPROCESS))
```

Used to determine if a widget is scheduled for configure

6.51.2.2 `#define CURRENT_H(o) EWL_OBJECT(o) → current.h`

Retrieve the current height of the object without insets or padding

6.51.2.3 `#define CURRENT_W(o) EWL_OBJECT(o) → current.w`

Retrieve the current width of the object without insets or padding

6.51.2.4 `#define CURRENT_X(o) EWL_OBJECT(o) → current.x`

Retrieve the current x position of the object

6.51.2.5 `#define CURRENT_Y(o) EWL_OBJECT(o) → current.y`

Retrieve the current y position of the object

6.51.2.6 `#define DESTROYED(o)`

Value:

```
(ewl_object_queued_has(EWL_OBJECT(o), \
                        EWL_FLAG_QUEUED_DSCHEDULED) \
 || ewl_object_queued_has(EWL_OBJECT(o), \
                        EWL_FLAG_QUEUED_DPROCESS))
```

Used to determine if a widget has been destroyed

6.51.2.7 `#define DISABLED(o) (ewl_object_state_has(EWL_OBJECT(o), EWL_FLAG_STATE_DISABLED))`

Used to determine if a widget is disabled

6.51.2.8 `#define EWL_OBJECT(object) ((Ewl_Object *) object)`

A typecast for accessing the inherited object fields.

6.51.2.9 `#define ewl_object_custom_h_set(o, h)`

Value:

```
{ \
    ewl_object_maximum_h_set(o, h); \
    ewl_object_minimum_h_set(o, h); \
    ewl_object_fill_policy_set(o, ewl_object_fill_policy_get(o) & \
                              ~(EWL_FLAG_FILL_VFILL | EWL_FLAG_FILL_VSHRINK)); \
}
```

Parameters:

o,: The object to work with

h,: The height to set A convenience method to set a custom height onto a widget

6.51.2.10 #define ewl_object_custom_size_set(o, w, h)**Value:**

```
{ \
    ewl_object_minimum_size_set(o, w, h); \
    ewl_object_maximum_size_set(o, w, h); \
    ewl_object_fill_policy_set(o, EWL_FLAG_FILL_NONE); \
}
```

Parameters:*o*,: The object to work with*w*,: The width to set*h*,: The height to set A convenience method to set a custom size onto a widget**6.51.2.11 #define ewl_object_custom_w_set(o, w)****Value:**

```
{ \
    ewl_object_maximum_w_set(o, w); \
    ewl_object_minimum_w_set(o, w); \
    ewl_object_fill_policy_set(o, ewl_object_fill_policy_get(o) & \
                                ~(EWL_FLAG_FILL_HFILL | EWL_FLAG_FILL_HSHRINK)); \
}
```

Parameters:*o*,: The object to work with*w*,: The width to set A convenience method to set a custom width onto a widget**6.51.2.12 #define ewl_object_flags_get(o, mask) (EWL_OBJECT(o) → flags & mask)**Retrieves the current setting of the object flags for *o*.**Parameters:***o*,: the parameter to retrieve the current value of object flags*mask*,: get only the flags specified in mask**Returns:**Returns the current setting of the object flags for *o*.**6.51.2.13 #define ewl_object_flags_has(o, check_flags, mask) (!! (EWL_OBJECT(o) → flags & ((check_flags) & mask)))**Determines if an object has the requested *flags* set.

Parameters:

o,: the object to check for a specified flags
check_flags,: the bitmask of flags to check on the object
mask,: get only the flags specified in mask

Returns:

Returns TRUE if any of the specified flags are set, FALSE otherwise.

6.51.2.14 `#define ewl_object_flags_has_all(o, check_flags, mask) ((EWL_OBJECT(o) → flags & ((check_flags) & mask)) == ((check_flags) & mask))`

Determines if an object has all of the requested *flags* set.

Parameters:

o,: the object to check for a specified flags
check_flags,: the bitmask of flags to check on the object
mask,: get only the flags specified in mask

Returns:

Returns TRUE if the specified flags are set, FALSE otherwise.

6.51.2.15 `#define ewl_object_in_tab_list_get(o) (ewl_object_flags_get(o, EWL_FLAG_PROPERTY_IN_TAB_LIST))`

Retrieves the current setting of the in tab list flag for *o*.

Parameters:

o,: the parameter to retrieve the current value of the in tab list flag

Returns:

Returns the current setting of the in tab list flag for *o*.

6.51.2.16 `#define ewl_object_in_tab_list_set(o, val)`

Value:

```
(val ? ewl_object_flags_add(o, EWL_FLAG_PROPERTY_IN_TAB_LIST, \
                             EWL_FLAGS_PROPERTY_MASK) : \
    ewl_object_flags_remove(o, EWL_FLAG_PROPERTY_IN_TAB_LIST, \
                             EWL_FLAGS_PROPERTY_MASK));
```

Changes the tab list flag value to match *val*.

Parameters:

o,: the object to change the in tab list

val,: a boolean indicating the value of the tab list flag

Returns:

Returns no value.

6.51.2.17 `#define EWL_OBJECT_MAX_SIZE (INT_MAX)`

The maximum possible size any object can receive.

6.51.2.18 `#define EWL_OBJECT_MIN_SIZE (1)`

The minimum possible size any object can receive.

6.51.2.19 `#define ewl_object_queued_add(o, queued) ewl_object_flags_add(o, queued, EWL_FLAGS_QUEUED_MASK)`

Parameters:

o,: The object to work with

queued,: Add the given queue flag to the object Adds the given queue flag *queued* to the object *o*

6.51.2.20 `#define ewl_object_queued_get(o, queued) ewl_object_flags_get(o, queued, EWL_FLAGS_QUEUED_MASK)`

Retrieve the value for the *queued* queue flag

6.51.2.21 `#define ewl_object_queued_has(o, queued) ewl_object_flags_has(o, queued, EWL_FLAGS_QUEUED_MASK)`

Determine if the *o* object has the *queued* flag set

6.51.2.22 `#define ewl_object_queued_remove(o, queued) ewl_object_flags_remove(o, queued, EWL_FLAGS_QUEUED_MASK)`

Remove the *queued* flag from the *o* object

6.51.2.23 `#define ewl_object_recursive_get(o) (ewl_object_flags_get(o, EWL_FLAG_PROPERTY_RECURSIVE))`

Retrieves the current setting of the recursive flag for *o*.

Parameters:

o,: the parameter to retrieve the current value of recursive flag

Returns:

Returns the current setting of the recursive flag for *o*.

6.51.2.24 `#define ewl_object_recursive_set(o, val)`**Value:**

```
(val ? ewl_object_flags_add(o, EWL_FLAG_PROPERTY_RECURSIVE, \
                             EWL_FLAGS_PROPERTY_MASK) : \
    ewl_object_flags_remove(o, EWL_FLAG_PROPERTY_RECURSIVE, \
                             EWL_FLAGS_PROPERTY_MASK));
```

Changes the recursive flag value to match *val*.

Parameters:

- o*,: the object to change the recursive flag
- val*,: a boolean indicating the value of the recursive flag

Returns:

Returns no value.

6.51.2.25 `#define ewl_object_state_add(o, state) ewl_object_flags_add(o, state, EWL_FLAGS_STATE_MASK)`**Parameters:**

- o*,: The object to work with
- state*,: The state to set into the object Adds the given state *state* to the object *o*

6.51.2.26 `#define ewl_object_state_get(o, state) ewl_object_flags_get(o, state, EWL_FLAGS_STATE_MASK)`**Parameters:**

- o*,: The object to work with
- state*,: The state to get Retrives the given state *state* from the object *o*

6.51.2.27 `#define ewl_object_state_has(o, state) ewl_object_flags_has(o, state, EWL_FLAGS_STATE_MASK)`**Parameters:**

- o*,: The object to check
- state*,: The state to check Checks if the given state *state* is set on the given object *o*

6.51.2.28 `#define ewl_object_state_remove(o, state) ewl_object_flags_remove(o, state, EWL_FLAGS_STATE_MASK)`**Parameters:**

- o*,: The object to work with
- state*,: The state to remove Removes the given state from the given *o* object

6.51.2.29 `#define ewl_object_toplevel_get(o) (ewl_object_flags_get(o, EWL_FLAG_PROPERTY_TOPLEVEL))`

Retrieves the current setting of the top level flag for *o*.

Parameters:

o,: the parameter to retrieve the current value of top level flag

Returns:

Returns the current setting of the top level flag for *o*.

6.51.2.30 `#define ewl_object_toplevel_set(o, val)`

Value:

```
(val ? ewl_object_flags_add(o, EWL_FLAG_PROPERTY_TOPLEVEL, \
                               EWL_FLAGS_PROPERTY_MASK) : \
    ewl_object_flags_remove(o, EWL_FLAG_PROPERTY_TOPLEVEL, \
                             EWL_FLAGS_PROPERTY_MASK));
```

Changes the top level flag value to match *val*.

Parameters:

o,: the object to change the top level flag

val,: a boolean indicating the value of the top level flag

Returns:

Returns no value.

6.51.2.31 `#define ewl_object_visible_add(o, visible) ewl_object_flags_add(o, visible, EWL_FLAGS_VISIBLE_MASK)`

Add the *visible* flag to the object *o*

6.51.2.32 `#define ewl_object_visible_get(o, visible) ewl_object_flags_get(o, visible, EWL_FLAGS_VISIBLE_MASK)`

Retrieves the *visible* flag from the object *o*

6.51.2.33 `#define ewl_object_visible_has(o, visible) ewl_object_flags_has(o, visible, EWL_FLAGS_VISIBLE_MASK)`

Check if the *visible* flag is set in the object *o*

6.51.2.34 `#define ewl_object_visible_remove(o, visible) ewl_object_flags_remove(o, visible, EWL_FLAGS_VISIBLE_MASK)`

Remove the *visible* flag from the object *o*

6.51.2.35 `#define HIDDEN(o) (!(EWL_OBJECT(o) → flags &
EWL_FLAG_VISIBLE_SHOWN))`

Used to determine if a widget is hidden.

6.51.2.36 `#define INSET_BOTTOM(o) EWL_OBJECT(o) → insets.b`

Retrieve the size of the bottom inset

6.51.2.37 `#define INSET_HORIZONTAL(o) (EWL_OBJECT(o) → insets.l +
EWL_OBJECT(o) → insets.r)`

Retrieve the total size of the horizontal insets for the object

6.51.2.38 `#define INSET_LEFT(o) EWL_OBJECT(o) → insets.l`

Retrieve the size of the left inset

6.51.2.39 `#define INSET_RIGHT(o) EWL_OBJECT(o) → insets.r`

Retrieve the size of the right inset

6.51.2.40 `#define INSET_TOP(o) EWL_OBJECT(o) → insets.t`

Retrieve the size of the top inset

6.51.2.41 `#define INSET_VERTICAL(o) (EWL_OBJECT(o) → insets.t +
EWL_OBJECT(o) → insets.b)`

Retrieve the total size of the vertical insets for the object

6.51.2.42 `#define MAXIMUM_H(o) EWL_OBJECT(o) → maximum.h`

Set the maximum height of the object

6.51.2.43 `#define MAXIMUM_W(o) EWL_OBJECT(o) → maximum.w`

Set the maximum width of the widget

6.51.2.44 `#define MINIMUM_H(o) EWL_OBJECT(o) → minimum.h`

Set the minimum height of the object

6.51.2.45 `#define MINIMUM_W(o) EWL_OBJECT(o) → minimum.w`

Set the minimum width of the object

6.51.2.46 `#define OBSCURED(o) (Ewl_Object(o) → flags & EWL_FLAG_VISIBLE_OBSCURED)`

Used to determine if a widget is marked as obscured.

6.51.2.47 `#define PADDING_BOTTOM(o) Ewl_Object(o) → pad.b`

Retrieve the size of the bottom pad

6.51.2.48 `#define PADDING_HORIZONTAL(o) (Ewl_Object(o) → pad.l + Ewl_Object(o) → pad.r)`

Retrieve the total size of the horizontal padding

6.51.2.49 `#define PADDING_LEFT(o) Ewl_Object(o) → pad.l`

Retrieve the size of the left pad

6.51.2.50 `#define PADDING_RIGHT(o) Ewl_Object(o) → pad.r`

Retrieve the size of the right pad

6.51.2.51 `#define PADDING_TOP(o) Ewl_Object(o) → pad.t`

Retrieve the size of the top pad

6.51.2.52 `#define PADDING_VERTICAL(o) (Ewl_Object(o) → pad.t + Ewl_Object(o) → pad.b)`

Retrieve the total size of the vertical padding

6.51.2.53 `#define PREFERRED_H(o) Ewl_Object(o) → preferred.h`

Set the preferred height of the widget

6.51.2.54 `#define PREFERRED_W(o) Ewl_Object(o) → preferred.w`

Set the preferred width of the widget

6.51.2.55 `#define REALIZED(o) (Ewl_Object(o) → flags & EWL_FLAG_VISIBLE_REALIZED)`

Used to test if a widget has been realized.

6.51.2.56 `#define RECURSIVE(o) (Ewl_Object(o) → flags & EWL_FLAG_PROPERTY_RECURSIVE)`

Used to test if a widget is recursive, aka. an `Ewl_Container` (p. 618)

6.51.2.57 `#define` `VISIBLE(o)` (`EWL_OBJECT(o)` \rightarrow `flags` &
`EWL_FLAG_VISIBLE_SHOWN`)

Used to test if a widget is visible.

6.51.3 Typedef Documentation

6.51.3.1 `typedef struct Ewl_Object Ewl_Object`

The base class inherited by all widgets. Provides methods for size and position.

6.51.4 Function Documentation

6.51.4.1 `unsigned int ewl_object_alignment_get (Ewl_Object * o)`

Retrieves the current setting of the object alignment for *o*.

Parameters:

o,: the parameter to retrieve the current value of object alignment

Returns:

Returns the current setting of the object alignment for *o*.

6.51.4.2 `void ewl_object_alignment_set (Ewl_Object * o, unsigned int align)`

Change the alignment of the specified object.

Parameters:

o,: the object to change alignment

align,: the new alignment for the object

Returns:

Returns no value.

Stores the new alignment value into the object for use when laying out the object.

6.51.4.3 `void ewl_object_current_geometry_get (Ewl_Object * o, int * x, int * y, int * w, int * h)`

Retrieve the size and position of object.

Parameters:

o,: the object to retrieve size and position

x,: a pointer to the integer to store the x coordinate

y,: a pointer to the integer to store the y coordinate

w,: a pointer to the integer to store the width

h,: a pointer to the integer to store the height

Returns:

Returns no value.

Examines *o* and stores its size and position into the integers pointed to by the parameters *x*, *y*, *w*, and *h*.

6.51.4.4 int ewl_object_current_h_get (Ewl_Object * *o*)

Get the current height of the object.

Parameters:

o,: the object to retrieve the current height

Returns:

Returns the current height of the object.

6.51.4.5 void ewl_object_current_size_get (Ewl_Object * *o*, int * *w*, int * *h*)

Retrieve the current size of an object.

Parameters:

o,: the object to retrieve size information

w,: a pointer to the integer to store the width of the object

h,: a pointer to the integer to store the height of the object

Returns:

Returns no value.

Stores the width and height of the object into *w* and *h* respectively.

6.51.4.6 int ewl_object_current_w_get (Ewl_Object * *o*)

Get the current width of the object.

Parameters:

o,: the object to retrieve the current width

Returns:

Returns the current width of the object.

6.51.4.7 int ewl_object_current_x_get (Ewl_Object * *o*)

get the current x position of the object

Parameters:

o,: the object to retrieve the current x position

Returns:

Returns the current x position of the object *o*.

6.51.4.8 int ewl_object_current_y_get (Ewl_Object * *o*)

Get the current y position of the object.

Parameters:

o,: the object to retrieve the current y position

Returns:

Returns the current y position of the object.

6.51.4.9 unsigned int ewl_object_fill_policy_get (Ewl_Object * *o*)

Retrieves the current setting of the object fill policy for *o*.

Parameters:

o,: the parameter to retrieve the current value of object fill policy

Returns:

Returns the current setting of the object fill policy for *o*.

6.51.4.10 void ewl_object_fill_policy_set (Ewl_Object * *o*, unsigned int *fill*)

Change the fill policy of the specified object.

Parameters:

o,: the object to change fill policy

fill,: the new fill policy for the object

Returns:

Returns no value.

See also:

Enums (p. 144)

Stores the new fill policy value into the object for use when laying out the object.

6.51.4.11 void ewl_object_flags_add (Ewl_Object * *o*, unsigned int *flags*, unsigned int *mask*)

Add the set of flags specified in *flags* to *o*.

Parameters:

o,: the object to set the specified object flags

flags,: a bitmask of new flags to be set in the object

mask,: a bitmask limiting added flags to a certain set

Returns:

Returns no value.

6.51.4.12 void ewl_object_flags_remove (Ewl_Object * *o*, unsigned int *flags*, unsigned int *mask*)

Removes the set of state flags specified in *flags* from *o*.

Parameters:

o,: the object to remove specified state flags

flags,: a bitmask of flags to be removed from the object

mask,: a bitmask limiting removed flags to a certain set

Returns:

Returns no value.

6.51.4.13 void ewl_object_geometry_request (Ewl_Object * *o*, int *x*, int *y*, int *w*, int *h*)

Request a new geometry be applied to an object.

Parameters:

o,: the object to request a new size

x,: the x coordinate to request be applied to the object

y,: the y coordinate to request be applied to the object

w,: the width to request be applied to the object

h,: the height to request be applied to the object

Returns:

Returns no value.

The given geometry is requested to be applied to the object. This is the usual method for requesting a new geometry for an object.

6.51.4.14 void ewl_object_h_request (Ewl_Object * *o*, int *h*)

Request a new width for an object.

Parameters:

o,: the object to request a new height

h,: the new height to be applied to the object

Returns:

Returns no value.

The given height is stored to be applied to the object *o* at a later time.

6.51.4.15 int ewl_object_init (Ewl_Object * *o*)**6.51.4.16 int ewl_object_insets_bottom_get (Ewl_Object * *o*)**

Get the bottom insets for the object.

Parameters:

o,: the ewl object to retrieve the bottom insets

Returns:

Returns the bottom insets for the object.

6.51.4.17 void ewl_object_insets_get (Ewl_Object * *o*, int * *l*, int * *r*, int * *t*, int * *b*)

Retrieve the edge insets of an object.

Parameters:

o,: the object to retrieve insets

l,: a pointer to an integer that receives the insets of the left side

r,: a pointer to an integer that receives the insets of the right side

t,: a pointer to an integer that receives the insets of the top side

b,: a pointer to an integer that receives the insets of the bottom side

Returns:

Returns no value.

Stores the edge insets of the object *o* into the integer pointer parameters passed in.

6.51.4.18 int ewl_object_insets_left_get (Ewl_Object * *o*)

Get the left insets for the object.

Parameters:

o,: the ewl object to retrieve the left insets

Returns:

Returns the left insets for the object.

6.51.4.19 int ewl_object_insets_right_get (Ewl_Object * *o*)

Get the right insets for the object.

Parameters:

o,: the ewl object to retrieve the right insets

Returns:

Returns the right insets for the object.

6.51.4.20 void ewl_object_insets_set (Ewl_Object * *o*, int *l*, int *r*, int *t*, int *b*)

Set the insets around an objects edges.

Parameters:

o,: the object to change the insets
l,: the new insets along the left side of the object
r,: the new insets along the right side of the object
t,: the new insets along the top side of the object
b,: the new insets along the bottom side of the object

Returns:

Returns no value.

Stores the values of *l*, *r*, *t* and *b* into the object to be used for distancing it's edges from other widgets when laying out.

6.51.4.21 int ewl_object_insets_top_get (Ewl_Object * *o*)

Get the top insets for the object.

Parameters:

o,: the ewl object to retrieve the top insets

Returns:

Returns the top insets for the object.

6.51.4.22 `int ewl_object_maximum_h_get (Ewl_Object * o)`

Get the maximum height of an object.

Parameters:

o,: the object to get the maximum height

Returns:

Returns the maximum height of the object.

6.51.4.23 `void ewl_object_maximum_h_set (Ewl_Object * o, int h)`

Set the minimum height of an object.

Parameters:

o,: the object to change the maximum height

h,: the new maximum height

Returns:

Returns no value.

Sets the maximum height of the object *o* to *h*. If the current height or minimum width are less than the new maximum, they are set to the new maximum height.

6.51.4.24 `void ewl_object_maximum_size_get (Ewl_Object * o, int * w, int * h)`

Retrieve the minimum dimensions of an object.

Parameters:

o,: the object to retrieve the maximum dimensions

w,: a pointer to an integer to store the maximum width

h,: a pointer to an integer to store the maximum height

Returns:

Returns no value.

Stores the maximum height and width of the object into the integers pointed to by *w* and *h* respectively.

6.51.4.25 `void ewl_object_maximum_size_set (Ewl_Object * o, int w, int h)`

Set the maximum size of an object.

Parameters:

o,: the object to change the maximum size

w,: the new maximum width

h,: the new maximum height

Returns:

Returns no value.

Sets the maximum size of the object *o* to *w* x *h*. If the current size or minimum size are less than the new maximum, they are set to the new maximum size.

6.51.4.26 int ewl_object_maximum_w_get (Ewl_Object * o)

Get the maximum width of an object.

Parameters:

o,: the object to get the maximum width

Returns:

Returns the maximum width of the object.

6.51.4.27 void ewl_object_maximum_w_set (Ewl_Object * o, int w)

Set the minimum width of an object.

Parameters:

o,: the object to change the maximum width

w,: the new maximum width

Returns:

Returns no value.

Sets the maximum width of the object *o* to *w*. If the current width or minimum width are less than the new maximum, they are set to the new maximum width.

6.51.4.28 int ewl_object_minimum_h_get (Ewl_Object * o)

Get the minimum height of an object.

Parameters:

o,: the object to get the minimum height

Returns:

Returns the minimum height of the object.

6.51.4.29 void ewl_object_minimum_h_set (Ewl_Object * *o*, int *h*)

Set the minimum height of an object.

Parameters:

o,: the object to change the minimum height

h,: the new minimum height

Returns:

Returns no value. Sets the minimum height of the object *o* to *h*. If the current height or maximum height are less than the new minimum, they are set to the new minimum height.

6.51.4.30 void ewl_object_minimum_size_get (Ewl_Object * *o*, int * *w*, int * *h*)

Retrieve the minimum dimensions of an object.

Parameters:

o,: the object to retrieve the minimum dimensions

w,: a pointer to an integer to store the minimum width

h,: a pointer to an integer to store the minimum height

Returns:

Returns no value.

Stores the minimum height and width of object *o* into the integers pointed to by *w* and *h* respectively.

6.51.4.31 void ewl_object_minimum_size_set (Ewl_Object * *o*, int *w*, int *h*)

Set the minimum size of an object.

Parameters:

o,: the object to change the minimum size

w,: the new minimum width

h,: the new minimum height

Returns:

Returns no value.

Sets the minimum size of the object *o* to *w* x *h*. If the current size or maximum size are less than the new minimum, they are set to the new minimum size.

6.51.4.32 int ewl_object_minimum_w_get (Ewl_Object * *o*)

Get the minimum width of an object.

Parameters:

o,: the object to get the minimum width

Returns:

Returns the minimum width of the object *o*.

6.51.4.33 void ewl_object_minimum_w_set (Ewl_Object * *o*, int *w*)

Set the minimum width of an object.

Parameters:

o,: the object to change the minimum width

w,: the new minimum width

Returns:

Returns no value.

Sets the minimum width of the object *o* to *w*. If the current width or maximum width are less than the new minimum, they are set to the new minimum width.

6.51.4.34 int ewl_object_padding_bottom_get (Ewl_Object * *o*)

Get the bottom padding for the object.

Parameters:

o,: the ewl object to retrieve the bottom padding

Returns:

Returns the bottom padding for the object.

6.51.4.35 void ewl_object_padding_get (Ewl_Object * *o*, int * *l*, int * *r*, int * *t*, int * *b*)

Retrieve the edge padding of an object.

Parameters:

o,: the object to retrieve padding

l,: a pointer to an integer that receives padding of the left side

r,: a pointer to an integer that receives padding of the right side

t,: a pointer to an integer that receives padding of the top side

b,: a pointer to an integer that receives padding of the bottom side

Returns:

Returns no value.

Stores the edge padding of the object *o* into the integer pointer parameters passed in.

6.51.4.36 int ewl_object_padding_left_get (Ewl_Object * o)

Get the left padding for the object.

Parameters:

o,: the ewl object to retrieve the left padding

Returns:

Returns the left padding for the object.

6.51.4.37 int ewl_object_padding_right_get (Ewl_Object * o)

Get the right padding for the object.

Parameters:

o,: the ewl object to retrieve the right padding

Returns:

Returns the right padding for the object.

6.51.4.38 void ewl_object_padding_set (Ewl_Object * o, int l, int r, int t, int b)

Set the padding around an objects edges.

Parameters:

o,: the object to change the padding

l,: the new padding along the left side of the object

r,: the new padding along the right side of the object

t,: the new padding along the top side of the object

b,: the new padding along the bottom side of the object

Returns:

Returns no value.

Stores the values of *l*, *r*, *t* and *b* into the object to be used for distancing it's edges from other widgets when laying out.

6.51.4.39 int ewl_object_padding_top_get (Ewl_Object * *o*)

Get the top padding for the object.

Parameters:

o,: the ewl object to retrieve the top padding

Returns:

Returns the top padding for the object.

6.51.4.40 void ewl_object_place (Ewl_Object * *o*, int *x*, int *y*, int *w*, int *h*)

Assign a specific area to an object.

Parameters:

o,: the object to place

x,: the x coordinate of the available area

y,: the y coordinate of the available area

w,: the width of the available area

h,: the height of the available area

Returns:

Returns no value.

Attempts to fill the object to the specified area, aligns the object within that area.

6.51.4.41 void ewl_object_position_request (Ewl_Object * *o*, int *x*, int *y*)

Request a new position be applied to an object.

Parameters:

o,: the object to request a new size

x,: the x coordinate to request be applied to the object

y,: the y coordinate to request be applied to the object

Returns:

Returns no value.

The given position is requested to be applied to the object *o* at a later time. This is the usual method for requesting a new position for an object.

6.51.4.42 `int ewl_object_preferred_h_get (Ewl_Object * o)`

Get the preferred height of the object.

Parameters:

o,: the object to retrieve the preferred height

Returns:

Returns the preferred height of the object.

6.51.4.43 `int ewl_object_preferred_inner_h_get (Ewl_Object * o)`

Get the preferred height of the object.

Parameters:

o,: the object to retrieve the preferred height

Returns:

Returns the preferred height of the object.

6.51.4.44 `void ewl_object_preferred_inner_h_set (Ewl_Object * o, int h)`

Set the preferred height of the object.

Parameters:

o,: the object to change preferred height

h,: the value to use as the preferred height

Returns:

Returns no value.

Sets the preferred of *o* height to *w* or as close as possible according to the bounds.

6.51.4.45 `void ewl_object_preferred_inner_size_get (Ewl_Object * o, int * w, int * h)`

Retrieve the preferred size of an object.

Parameters:

o,: the object to retrieve size information

w,: a pointer to the integer to store the width of the object

h,: a pointer to the integer to store the height of the object

Returns:

Returns no value.

Stores the width and height of object *o* into *w* and *h* respectively, without the insets and padding added.

6.51.4.46 `void ewl_object_preferred_inner_size_set (Ewl_Object * o, int w, int h)`

Set the preferred size of the object.

Parameters:

o,: the object to change size
w,: the new width of the object
h,: the new height of the object

Returns:

Returns no value.

The dimensions of the object *o* are set to the values of the parameters *w*, and *h* unless these values are greater than the objects maximum value or smaller than the objects minimum value. If they are outside these bounds, the size is not altered.

6.51.4.47 `int ewl_object_preferred_inner_w_get (Ewl_Object * o)`

Get the preferred width of the object, ignoring padding and insets.

Parameters:

o,: the object to retrieve the preferred width

Returns:

Returns the preferred width of the object *o*.

6.51.4.48 `void ewl_object_preferred_inner_w_set (Ewl_Object * o, int w)`

Set the preferred width of the object.

Parameters:

o,: the object to change preferred width
w,: the value to use as the preferred width

Returns:

Returns no value.

Sets the preferred of *o* width to *w* or as close as possible according to the bounds.

6.51.4.49 `void ewl_object_preferred_size_get (Ewl_Object * o, int * w, int * h)`

Retrieve the preferred size of an object.

Parameters:

o,: the object to retrieve size information

w,: a pointer to the integer to store the width of the object

h,: a pointer to the integer to store the height of the object

Returns:

Returns no value.

Stores the width and height of object *o* into *w* and *h* respectively.

6.51.4.50 int ewl_object_preferred_w_get (Ewl_Object * *o*)

Get the preferred width of the object.

Parameters:

o,: the object to retrieve the preferred width

Returns:

Returns the preferred width of the object *o*.

6.51.4.51 void ewl_object_size_request (Ewl_Object * *o*, int *w*, int *h*)

Request a new size be applied to an object.

Parameters:

o,: the object to request a new size

w,: the width to request be applied to the object

h,: the height to request be applied to the object

Returns:

Returns no value.

The given size is requested to be applied to the object *o* at a later time. This is the usual method for requesting a new size for an object.

6.51.4.52 void ewl_object_w_request (Ewl_Object * *o*, int *w*)

Request a new width for an object.

Parameters:

o,: the object to request a new width

w,: the new width to be applied to the object

Returns:

Returns no value.

The given width is stored to be applied to the object *o* at a later time.

6.51.4.53 void ewl_object_x_request (Ewl_Object * *o*, int *x*)

Request a new x position for an object.

Parameters:

o,: the object to request a new x position

x,: the new x coordinate to be applied to the object

Returns:

Returns no value.

The given x coordinate is stored to be applied to the object *o* at a later time.

6.51.4.54 void ewl_object_y_request (Ewl_Object * *o*, int *y*)

Request a new y position for an object.

Parameters:

o,: the object to request a new y position

y,: the new y coordinate to be applied to the object

Returns:

Returns no value.

The given y coordinate is stored to be applied to the object *o* at a later time.

6.52 Ewl_Overlay: A Container for Displaying on a canvas

6.52.1 Detailed Description

Defines the **Ewl_Overlay** (p. 712) class to provide EWL with the ability to work with an evas.

Remarks:

Inherits from **Ewl_Container** (p. 618).

Data Structures

- struct **Ewl_Overlay**

*The class inheriting from **Ewl_Container** (p. 618) that acts as a top level widget for interacting with the evas.*

Defines

- #define **EWL_OVERLAY**(widget) ((**Ewl_Overlay** *) widget)
*Typecast a pointer to an **Ewl_Overlay** (p. 712) pointer.*
- #define **EWL_OVERLAY_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_OVERLAY_TYPE**))
- #define **EWL_OVERLAY_TYPE** "overlay"

Typedefs

- typedef **Ewl_Overlay** **Ewl_Overlay**

Functions

- void **ewl_overlay_cb_child_resize** (**Ewl_Container** *c, **Ewl_Widget** *w, int size, **Ewl_Orientation** o)
- void **ewl_overlay_cb_child_show** (**Ewl_Container** *emb, **Ewl_Widget** *child)
- void **ewl_overlay_cb_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- int **ewl_overlay_init** (**Ewl_Overlay** *win)
initialize a overlay to default values and callbacks
- **Ewl_Widget** * **ewl_overlay_new** (void)
Allocate and initialize a new overlay container.

6.52.2 Define Documentation

6.52.2.1 #define **EWL_OVERLAY**(widget) ((**Ewl_Overlay** *) widget)

Typecast a pointer to an **Ewl_Overlay** (p. 712) pointer.

6.52.2.2 `#define EWL_OVERLAY_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_OVERLAY_TYPE))`

Returns TRUE if the widget is an **Ewl_Overlay** (p. 712), FALSE otherwise

6.52.2.3 `#define EWL_OVERLAY_TYPE "overlay"`

The type name for the **Ewl_Overlay** (p. 712) widget

6.52.3 Typedef Documentation

6.52.3.1 `typedef struct Ewl_Overlay Ewl_Overlay`

The overlay structure is mostly a container for holding widgets and a wrapper evas smart object.

6.52.4 Function Documentation

6.52.4.1 `void ewl_overlay_cb_child_resize (Ewl_Container * c, Ewl_Widget * w, int size, Ewl_Orientation o)`

6.52.4.2 `void ewl_overlay_cb_child_show (Ewl_Container * emb, Ewl_Widget * child)`

6.52.4.3 `void ewl_overlay_cb_configure (Ewl_Widget * w, void * ev_data, void * user_data)`

6.52.4.4 `int ewl_overlay_init (Ewl_Overlay * w)`

initialize a overlay to default values and callbacks

Parameters:

w,: the overlay to be initialized to default values and callbacks

Returns:

Returns TRUE or FALSE depending on if initialization succeeds.

Sets the values and callbacks of a overlay *w* to their defaults.

6.52.4.5 `Ewl_Widget* ewl_overlay_new (void)`

Allocate and initialize a new overlay container.

Returns:

Returns a new overlay container on success, or NULL on failure.

6.53 Ewl_Paned: A paned widget

6.53.1 Detailed Description

Provides a widget to have a resizable paned container

Remarks:

Inherits from **Ewl_Container** (p. 618).

6.53.2 Information

The **Ewl_Paned** (p. 715) widget is designed to hold multiple panes if needed. If you want to have three panes or four panes just add the panes and it will add the grabbers as needed. The paned widget is an **Ewl_Container** (p. 618) so to add panes you just use the container functions to append, prepend and insert as needed.

6.53.3 Initial Layout

If you need to set the initial sizes of the items in the grabber then you will need to set their preferred size. If, after the paned is already displayed, you need to change the size it will use the current size of the pane.

Data Structures

- struct **Ewl_Paned**

*Inherits from **Ewl_Container** (p. 618) and extends to provide the paned widget.*

- struct **Ewl_Paned_Grabber**

*Inherits from **Ewl_Separator** (p. 740) and extends to facilitate the paned grabber.*

Defines

- `#define EWL_PANED(paned) ((Ewl_Paned *) paned)`
- `#define EWL_PANED_GRABBER(g) ((Ewl_Paned_Grabber *)g)`
- `#define EWL_PANED_GRABBER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_PANED_GRABBER_TYPE))`
- `#define EWL_PANED_GRABBER_TYPE "paned_grabber"`
- `#define EWL_PANED_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_PANED_TYPE))`
- `#define EWL_PANED_TYPE "paned"`

Typedefs

- `typedef Ewl_Paned Ewl_Paned`
- `typedef Ewl_Paned_Grabber Ewl_Paned_Grabber`

Functions

- **Ewl_Widget * ewl_hpaned_new** (void)
Allocate and initialize a new paned widget with horizontal orientation.
- void **ewl_paned_cb_child_add** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_paned_cb_child_hide** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_paned_cb_child_remove** (Ewl_Container *c, Ewl_Widget *w, int idx)
- void **ewl_paned_cb_child_show** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_paned_cb_configure** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_paned_cb_destroy** (Ewl_Widget *w, void *ev, void *data)
- int **ewl_paned_grabber_init** (Ewl_Paned_Grabber *g)
- **Ewl_Widget * ewl_paned_grabber_new** (void)
- **Ewl_Orientation ewl_paned_grabber_orientation_get** (Ewl_Paned_Grabber *g)
- void **ewl_paned_grabber_paned_orientation_set** (Ewl_Paned_Grabber *g, Ewl_Orientation o)
- void **ewl_paned_grabber_show_cursor_for** (Ewl_Paned_Grabber *g, unsigned int dir)
- int **ewl_paned_init** (Ewl_Paned *p)
Initialize a new paned widget to default values.
- **Ewl_Widget * ewl_paned_new** (void)
Allocate and initialize a new paned widget.
- **Ewl_Orientation ewl_paned_orientation_get** (Ewl_Paned *p)
Get the orientation of the paned widget.
- void **ewl_paned_orientation_set** (Ewl_Paned *p, Ewl_Orientation o)
Set the orientation of the paned.
- **Ewl_Widget * ewl_vpaned_new** (void)
Allocate and initialize a new paned widget with vertical orientation.

6.53.4 Define Documentation

6.53.4.1 #define EWL_PANED(paned) ((Ewl_Paned *) paned)

Typecasts a pointer to a **Ewl_Paned** (p. 715) pointer

6.53.4.2 #define EWL_PANED_GRABBER(g) ((Ewl_Paned_Grabber *)g)

Typecasts a pointer to an **Ewl_Paned_Grabber** (p. 717) pointer

6.53.4.3 #define EWL_PANED_GRABBER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_PANED_GRABBER_TYPE))

Returns TRUE if the widget is an **Ewl_Paned_Grabber** (p. 717), FALSE otherwise

6.53.4.4 `#define EWL_PANED_GRABBER_TYPE "paned_grabber"`

The type name for the **Ewl_Paned_Grabber** (p. 717) widget

6.53.4.5 `#define EWL_PANED_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_PANED_TYPE))`

Returns TRUE if the widget is an **Ewl_Paned** (p. 715), FALSE otherwise

6.53.4.6 `#define EWL_PANED_TYPE "paned"`

The type name for the **Ewl_Paned** (p. 715) widget

6.53.5 Typedef Documentation**6.53.5.1** `typedef struct Ewl_Paned Ewl_Paned`

The **Ewl_Paned** (p. 715) widget

6.53.5.2 `typedef struct Ewl_Paned_Grabber Ewl_Paned_Grabber`

Typedef for the **Ewl_Paned_Grabber** (p. 717) struct

6.53.6 Function Documentation**6.53.6.1** `Ewl_Widget* ewl_hpaned_new (void)`

Allocate and initialize a new paned widget with horizontal orientation.

Returns:

Returns NULL on failure, or a pointer to a new paned widget on success.

- 6.53.6.2 void ewl_paned_cb_child_add (Ewl_Container * *c*, Ewl_Widget * *w*)
- 6.53.6.3 void ewl_paned_cb_child_hide (Ewl_Container * *c*, Ewl_Widget * *w*)
- 6.53.6.4 void ewl_paned_cb_child_remove (Ewl_Container * *c*, Ewl_Widget * *w*, int *idx*)
- 6.53.6.5 void ewl_paned_cb_child_show (Ewl_Container * *c*, Ewl_Widget * *w*)
- 6.53.6.6 void ewl_paned_cb_configure (Ewl_Widget * *w*, void * *ev*, void * *data*)
- 6.53.6.7 void ewl_paned_cb_destroy (Ewl_Widget * *w*, void * *ev*, void * *data*)
- 6.53.6.8 int ewl_paned_grabber_init (Ewl_Paned_Grabber * *g*)
- 6.53.6.9 Ewl_Widget* ewl_paned_grabber_new (void)
- 6.53.6.10 Ewl_Orientation ewl_paned_grabber_orientation_get (Ewl_Paned_Grabber * *g*)
- 6.53.6.11 void ewl_paned_grabber_paned_orientation_set (Ewl_Paned_Grabber * *g*, Ewl_Orientation *o*)
- 6.53.6.12 void ewl_paned_grabber_show_cursor_for (Ewl_Paned_Grabber * *g*, unsigned int *dir*)
- 6.53.6.13 int ewl_paned_init (Ewl_Paned * *p*)

Initialize a new paned widget to default values.

Parameters:

p,: the paned widget to initialize

Returns:

Returns 1 on success or 0 on failure

- 6.53.6.14 Ewl_Widget* ewl_paned_new (void)

Allocate and initialize a new paned widget.

Returns:

Returns NULL on failure, or a pointer to a new paned widget on success.

- 6.53.6.15 Ewl_Orientation ewl_paned_orientation_get (Ewl_Paned * *p*)

Get the orientation of the paned widget.

Parameters:

p,: The Ewl_Paned (p. 715) to get the orientation from

Returns:

Returns the Ewl_Orientation currently set on the paned

6.53.6.16 void ewl_paned_orientation_set (Ewl_Paned * *p*, Ewl_Orientation *o*)

Set the orientation of the paned.

Parameters:

p,: The paned to set the orientation on

o,: The Ewl_Orientation to set

Returns:

Returns no value.

6.53.6.17 Ewl_Widget* ewl_vpaned_new (void)

Allocate and initialize a new paned widget with vertical orientation.

Returns:

Returns NULL on failure, or a pointer to a new paned widget on success.

6.54 Ewl_Password: A Single Line Text Password Widget

6.54.1 Detailed Description

Defines the **Ewl_Password** (p. 719) class to allow for single line obscured text.

Remarks:

Inherits from **Ewl_Entry** (p. 635).

6.54.2 Tutorial

The password extends a basic **Ewl_Entry** (p. 635) to provide an obscured view of the text entered.

```
Ewl_Widget *password;
password = ewl_password_new();
ewl_widget_show(password);
```

Manipulation of the password widget is generally done through the **Ewl_Entry** (p. 635) API since the password widget inherits from the entry. The first point it diverges from the entry API is for retrieving the current string in the widget. This is necessary because the entry API returns the text that is hiding the actual contents entered.

```
void ewl_password_text_set(Ewl_Password *e, const char *t);
char *ewl_password_text_get(Ewl_Password *e);
```

The character used to hide the text of the password is also configurable at runtime, the default obscuring character is '*'.

```
void ewl_password_obscure_set(Ewl_Password *e, char o);
char ewl_password_obscure_get(Ewl_Password *e);
```

Data Structures

- struct **Ewl_Password**

*Inherits from the **Ewl_Entry** (p. 635) and extends it to obscure the text.*

Defines

- #define **EWL_PASSWORD**(password) ((**Ewl_Password** *) password)
- #define **EWL_PASSWORD_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_PASSWORD_TYPE**))
- #define **EWL_PASSWORD_TYPE** "password"

Typedefs

- typedef **Ewl_Password** **Ewl_Password**

Functions

- void **ewl_password_cb_destroy** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_password_cb_key_down** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- int **ewl_password_init** (**Ewl_Password** *e)
Initialize an password widget to default values.
- **Ewl_Widget** * **ewl_password_new** (void)
Allocate and initialize a new password widget.
- char **ewl_password_obscure_get** (**Ewl_Password** *e)
Retrieves the character used to obscure the text for a password.
- void **ewl_password_obscure_set** (**Ewl_Password** *e, char o)
Sets the character used to obscure the text for a password.
- char * **ewl_password_text_get** (**Ewl_Password** *e)
Get the text from an password widget.
- void **ewl_password_text_set** (**Ewl_Password** *e, const char *t)
Set the text for an password widget.

6.54.3 Define Documentation

6.54.3.1 `#define EWL_PASSWORD(password) ((Ewl_Password *) password)`

Typecasts a pointer to an **Ewl_Password** (p. 719) pointer.

6.54.3.2 `#define EWL_PASSWORD_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_PASSWORD_TYPE))`

Returns TRUE if the widget is an **Ewl_Password** (p. 719), FALSE otherwise

6.54.3.3 `#define EWL_PASSWORD_TYPE "password"`

The type name for the **Ewl_Password** (p. 719) widget

6.54.4 Typedef Documentation

6.54.4.1 `typedef struct Ewl_Password Ewl_Password`

Inherits from the **Ewl_Widget** (p. 785) and provides single line editable text.

6.54.5 Function Documentation

6.54.5.1 void ewl_password_cb_destroy (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.54.5.2 void ewl_password_cb_key_down (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.54.5.3 int ewl_password_init (Ewl_Password * *e*)

Initialize an password widget to default values.

Parameters:

e,: the password widget to initialize

Returns:

Returns no value.

Initializes the password widget *e* to it's default values and callbacks.

6.54.5.4 Ewl_Widget* ewl_password_new (void)

Allocate and initialize a new password widget.

Returns:

Returns a new password widget on success, NULL on failure.

6.54.5.5 char ewl_password_obscure_get (Ewl_Password * *e*)

Retrieves the character used to obscure the text for a password.

Parameters:

e,: the password to retrieve the obscuring character

Returns:

Returns the character value of the obscuring character.

6.54.5.6 void ewl_password_obscure_set (Ewl_Password * *e*, char *o*)

Sets the character used to obscure the text for a password.

Parameters:

e,: set the obscuring character for a passwords text

o,: the character to obscure the password characters

Returns:

Returns no value.

6.54.5.7 `char* ewl_password_text_get (Ewl_Password * e)`

Get the text from an password widget.

Parameters:

e,: the password widget to retrieve the text

Returns:

Returns the password text on success, NULL on failure.

6.54.5.8 `void ewl_password_text_set (Ewl_Password * e, const char * t)`

Set the text for an password widget.

Parameters:

e,: the password widget to change the text

t,: the text to set for the password widget

Returns:

Returns no value.

Change the text of the password widget *e* to the string *t*.

6.55 Ewl_Popup: A override window for the convenience use

6.55.1 Detailed Description

Defines the **Ewl_Popup** (p. 721) class used for override window that need to be placed with some kind of smartness

Remarks:

Inherits from **Ewl_Window** (p. 788).

Data Structures

- struct **Ewl_Popup**

*Inherits from **Ewl_Window** (p. 788) and extends to provide a popup widget.*

Defines

- #define **EWL_POPUP**(popup) ((**Ewl_Popup** *) popup)
- #define **EWL_POPUP_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_POPUP_TYPE**))
- #define **EWL_POPUP_TYPE** "popup"

Typedefs

- typedef **Ewl_Popup** **Ewl_Popup**

Functions

- void **ewl_popup_cb_child_resize** (**Ewl_Container** *c, **Ewl_Widget** *w, int size, **Ewl_Orientation** o)
- void **ewl_popup_cb_child_show** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_popup_cb_destroy** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_popup_cb_follow_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_popup_cb_follow_destroy** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_popup_cb_show** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- int **ewl_popup_fit_to_follow_get** (**Ewl_Popup** *p)
- void **ewl_popup_fit_to_follow_set** (**Ewl_Popup** *p, int fit)
- **Ewl_Widget** * **ewl_popup_follow_get** (**Ewl_Popup** *p)
- void **ewl_popup_follow_set** (**Ewl_Popup** *p, **Ewl_Widget** *w)
Set the follow widget of the popup.
- int **ewl_popup_init** (**Ewl_Popup** *p)
Initialize the popup and inherited fields.

- void **ewl_popup_mouse_position_set** (**Ewl_Popup** *p, int x, int y)

This is to set the mouse coordinates to the popup, where it get placed relative to.

- **Ewl_Widget** * **ewl_popup_new** (void)

Allocate a new popup widget.

- void **ewl_popup_offset_set** (**Ewl_Popup** *p, int x, int y)

This is to set the offset to where the popup will be placed.

- **Ewl_Popup_Type** **ewl_popup_type_get** (**Ewl_Popup** *p)

This is to get the type of the popup. The type defines how the popup will handle the positioning.

- void **ewl_popup_type_set** (**Ewl_Popup** *p, **Ewl_Popup_Type** type)

This is to set the type of the popup. The type defines how the popup will handle the positioning.

6.55.2 Define Documentation

6.55.2.1 #define **EWL_POPUP**(popup) ((**Ewl_Popup** *) popup)

Typecast a pointer to an **Ewl_Popup** (p. 721) pointer.

6.55.2.2 #define **EWL_POPUP_IS**(w) (**ewl_widget_type_is**(**EWL_WIDGET**(w), **EWL_POPUP_TYPE**))

Returns TRUE if the widget is an **Ewl_Popup** (p. 721), FALSE otherwise

6.55.2.3 #define **EWL_POPUP_TYPE** "popup"

The type name for the **Ewl_Popup** (p. 721) widget

6.55.3 Typedef Documentation

6.55.3.1 typedef struct **Ewl_Popup** **Ewl_Popup**

Provides an **Ewl_Widget** (p. 785) to simply act with override windows

6.55.4 Function Documentation

6.55.4.1 void ewl_popup_cb_child_resize (Ewl_Container * *c*, Ewl_Widget * *w*, int *size*, Ewl_Orientation *o*)

6.55.4.2 void ewl_popup_cb_child_show (Ewl_Container * *c*, Ewl_Widget * *w*)

6.55.4.3 void ewl_popup_cb_destroy (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.55.4.4 void ewl_popup_cb_follow_configure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.55.4.5 void ewl_popup_cb_follow_destroy (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.55.4.6 void ewl_popup_cb_show (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.55.4.7 int ewl_popup_fit_to_follow_get (Ewl_Popup * *p*)

Parameters:

p,: The popup

Returns:

Returns if the popup is set to fit the size of the follow

6.55.4.8 void ewl_popup_fit_to_follow_set (Ewl_Popup * *p*, int *fit*)

Parameters:

p,: The popup

fit,: TRUE or FALSE

Returns:

Returns no value

6.55.4.9 Ewl_Widget* ewl_popup_follow_get (Ewl_Popup * *p*)

Parameters:

p,: The popup to get the follow from

Returns:

Returns the follow of the popup

6.55.4.10 void ewl_popup_follow_set (Ewl_Popup * *p*, Ewl_Widget * *w*)

Set the follow widget of the popup.

Parameters:

p,: The popup to set the follow widget

w,: The widget to follow

Returns:

Returns no value

6.55.4.11 int ewl_popup_init (Ewl_Popup * *p*)

Initialize the popup and inherited fields.

Parameters:

p,: the popup to initialize

Returns:

Returns TRUE on success, FALSE on failure.

Clears the contents of the popup and stores the default values.

6.55.4.12 void ewl_popup_mouse_position_set (Ewl_Popup * *p*, int *x*, int *y*)

This is to set the mouse coordinates to the popup, where it get placed relative to.

Parameters:

p,: The popup to set the mouse position for

x,: the x coordinate

y,: the y coordinate

Returns:

Returns no value

6.55.4.13 Ewl_Widget* ewl_popup_new (void)

Allocate a new popup widget.

Returns:

Returns pointer to new popup widget on success, NULL on failure.

6.55.4.14 void ewl_popup_offset_set (Ewl_Popup * *p*, int *x*, int *y*)

This is to set the offset to where the popup will be placed.

Parameters:

p,: The popup to set the offset for

x,: the x offset

y,: the y offset

Returns:

Returns no value

6.55.4.15 Ewl_Popup_Type ewl_popup_type_get (Ewl_Popup * *p*)

This is to get the type of the popup. The type defines how the popup will handle the positioning.

Parameters:

p,: The popup to get the type from

Returns:

Returns the type of the popup

6.55.4.16 void ewl_popup_type_set (Ewl_Popup * *p*, Ewl_Popup_Type *type*)

This is to set the type of the popup. The type defines how the popup will handle the positioning.

Parameters:

p,: The popup to set the type for

type,: the new type of the popup

Returns:

Returns no value

6.56 Ewl_Progressbar: A progressbar from a value

6.56.1 Detailed Description

A progress bar widget

Remarks:

Inherits from **Ewl_Range** (p. 727).

Data Structures

- struct **Ewl_Progressbar**

*Inherits from **Ewl_Range** (p. 727) and extends to provide a progressbar widget.*

Defines

- #define **EWL_PROGRESSBAR**(progressbar) ((**Ewl_Progressbar** *) progressbar)
- #define **EWL_PROGRESSBAR_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_PROGRESSBAR_TYPE))
- #define **EWL_PROGRESSBAR_TYPE** "progressbar"

Typedefs

- typedef **Ewl_Progressbar** **Ewl_Progressbar**

Functions

- void **ewl_progressbar_cb_child_resize** (**Ewl_Container** *c, **Ewl_Widget** *w, int size, **Ewl_Orientation** o)
- void **ewl_progressbar_cb_child_show** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_progressbar_cb_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_progressbar_cb_value_changed** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_progressbar_custom_label_set** (**Ewl_Progressbar** *p, char *format_string)

Sets the given format string on the progressbar (lf of lf beers).

- int **ewl_progressbar_init** (**Ewl_Progressbar** *p)

Initialize the progressbar to some sane starting values.

- void **ewl_progressbar_label_hide** (**Ewl_Progressbar** *p)

Hides the given progressbars label.

- void **ewl_progressbar_label_set** (**Ewl_Progressbar** *p, char *label)

Sets the given text on the progressbar.

- void ewl_progressbar_label_show (Ewl_Progressbar *p)

Shows the given progressbars label.

- Ewl_Widget * ewl_progressbar_new (void)

Allocate and initialize a new progressbar.

6.56.2 Define Documentation

6.56.2.1 #define EWL_PROGRESSBAR(progressbar) ((Ewl_Progressbar *) progressbar)

Typecasts a pointer to an **Ewl_Progressbar** (p. 723) pointer.

6.56.2.2 #define EWL_PROGRESSBAR_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_PROGRESSBAR_TYPE))

Returns TRUE if the widget is an **Ewl_Progressbar** (p. 723), FALSE otherwise

6.56.2.3 #define EWL_PROGRESSBAR_TYPE "progressbar"

The type name for the **Ewl_Progressbar** (p. 723) widget

6.56.3 Typedef Documentation

6.56.3.1 typedef struct Ewl_Progressbar Ewl_Progressbar

The **Ewl_Progressbar** (p. 723) provides a statusbar from a given value

6.56.4 Function Documentation

6.56.4.1 void ewl_progressbar_cb_child_resize (Ewl_Container * c, Ewl_Widget * w, int size, Ewl_Orientation o)

6.56.4.2 void ewl_progressbar_cb_child_show (Ewl_Container * c, Ewl_Widget * w)

6.56.4.3 void ewl_progressbar_cb_configure (Ewl_Widget * w, void * ev_data, void * user_data)

6.56.4.4 void ewl_progressbar_cb_value_changed (Ewl_Widget * w, void * ev_data, void * user_data)

6.56.4.5 void ewl_progressbar_custom_label_set (Ewl_Progressbar * p, char * format_string)

Sets the given format string on the progressbar (lf of lf beers).

Parameters:

p,: the progressbars whose label will be changed
format_string,: format string for the new label

Returns:

Returns no value

6.56.4.6 int ewl_progressbar_init (Ewl_Progressbar * *p*)

Initialize the progressbar to some sane starting values.

Parameters:

p,: the progressbar to be initialized

Returns:

Returns TRUE on success, FALSE on failure.

6.56.4.7 void ewl_progressbar_label_hide (Ewl_Progressbar * *p*)

Hides the given progressbars label.

Parameters:

p,: the progressbars whose label will be hidden

Returns:

Returns no value

6.56.4.8 void ewl_progressbar_label_set (Ewl_Progressbar * *p*, char * *label*)

Sets the given text on the progressbar.

Parameters:

p,: the progressbars whose text will be changed
label,: the new label

Returns:

Returns no value

6.56.4.9 void ewl_progressbar_label_show (Ewl_Progressbar * *p*)

Shows the given progressbars label.

Parameters:

p,: the progressbars whose label will be shown

Returns:

Returns no value

6.56.4.10 Ewl_Widget* ewl_progressbar_new (void)

Allocate and initialize a new progressbar.

Returns:

Returns NULL on failure, or a pointer to the new progressbar on success.

6.57 Ewl_Radiobutton: A Radio Button Widget and Grouping System

6.57.1 Detailed Description

Provides for a simple radiobutton with label, and to group radio buttons for selecting a single option.

Remarks:

Inherits from **Ewl_Checkbutton** (p. 598).

Data Structures

- struct **Ewl_Radiobutton**

*Inherits from **Ewl_Checkbutton** (p. 598) and extends it to provide grouping buttons to limit to a single selection in a group at a given time.*

Defines

- #define **EWL_RADIOBUTTON**(button) ((**Ewl_Radiobutton** *) button)
- #define **ewl_radiobutton_checked_set**(r, c) ewl_checkbutton_checked_set(**EWL_CHECKBUTTON**(r), c)
- #define **EWL_RADIOBUTTON_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_RADIOBUTTON_TYPE**))
- #define **ewl_radiobutton_is_checked**(r) ewl_checkbutton_is_checked(**EWL_CHECKBUTTON**(r))
- #define **EWL_RADIOBUTTON_TYPE** "radiobutton"

Typedefs

- typedef **Ewl_Radiobutton** **Ewl_Radiobutton**

Functions

- void **ewl_radiobutton_cb_clicked** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_radiobutton_cb_destroy** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_radiobutton_chain_set** (**Ewl_Radiobutton** *w, **Ewl_Radiobutton** *c)
Attach the button to a chain of radio buttons.
- int **ewl_radiobutton_init** (**Ewl_Radiobutton** *cb)
Initialize the radio button fields and callbacks.
- **Ewl_Widget** * **ewl_radiobutton_new** (void)
Allocate and initialize a new radio button.

6.57.2 Define Documentation

6.57.2.1 #define EWL_RADIOBUTTON(button) ((Ewl_Radiobutton *) button)

Typecasts a pointer to an **Ewl_Radiobutton** (p. 725) pointer.

6.57.2.2 #define ewl_radiobutton_checked_set(r, c) ewl_checkbutton_checked_set(EWL_CHECKBUTTON(r), c)

Shortcut for setting the checked status on the inherited **Ewl_Checkbutton** (p. 598) fields.

6.57.2.3 #define EWL_RADIOBUTTON_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_RADIOBUTTON_TYPE))

Returns TRUE if the widget is an **Ewl_Radiobutton** (p. 725), FALSE otherwise

6.57.2.4 #define ewl_radiobutton_is_checked(r) ewl_checkbutton_is_checked(EWL_CHECKBUTTON(r))

Shortcut for checking the checked status on the inherited **Ewl_Checkbutton** (p. 598) fields.

6.57.2.5 #define EWL_RADIOBUTTON_TYPE "radiobutton"

The type name for the **Ewl_Radiobutton** (p. 725) widget

6.57.3 Typedef Documentation

6.57.3.1 typedef struct Ewl_Radiobutton Ewl_Radiobutton

The radio button provides a means for selecting a single item from a group of options.

6.57.4 Function Documentation

6.57.4.1 void ewl_radiobutton_cb_clicked (Ewl_Widget * w, void * ev_data, void * user_data)

6.57.4.2 void ewl_radiobutton_cb_destroy (Ewl_Widget * w, void * ev_data, void * user_data)

6.57.4.3 void ewl_radiobutton_chain_set (Ewl_Radiobutton * rb, Ewl_Radiobutton * crb)

Attach the button to a chain of radio buttons.

Parameters:

rb,: the radio button to be added to a chain of radio buttons

crb,: a radio button already in the chain of radio buttons

Returns:

Returns no value.

Associates *w* with the same chain as *c*, in order to ensure that only one radio button of that group is checked at any time.

6.57.4.4 int ewl_radiobutton_init (Ewl_Radiobutton * *rb*)

Initialize the radio button fields and callbacks.

Parameters:

rb,: the radio button to initialize

Returns:

Returns no value.

Sets internal fields of the radio button to default values and sets the label to the specified *label*.

6.57.4.5 Ewl_Widget* ewl_radiobutton_new (void)

Allocate and initialize a new radio button.

Returns:

Returns a pointer to new radio button on success, NULL on failure.

6.58 Ewl_Range: A Value Selector from a Range

6.58.1 Detailed Description

Defines an **Ewl_Widget** (p. 785) with a draggable button enclosed, used to select a value from a range.

Remarks:

Inherits from **Ewl_Container** (p. 618).

Data Structures

- struct **Ewl_Range**

*Inherits from **Ewl_Container** (p. 618) and extends to provide a basic range widget.*

Defines

- #define **EWL_RANGE**(range) ((**Ewl_Range** *) range)
- #define **EWL_RANGE_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_RANGE_TYPE**))
- #define **EWL_RANGE_TYPE** "range"

Typedefs

- typedef **Ewl_Range** **Ewl_Range**

Functions

- void **ewl_range_decrease** (**Ewl_Range** *r)
Decrease the value of a range by it's step size.
- void **ewl_range_increase** (**Ewl_Range** *r)
Increase the value of a range by it's step size.
- int **ewl_range_init** (**Ewl_Range** *r)
Initialize the range to some sane starting values.
- int **ewl_range_invert_get** (**Ewl_Range** *r)
Retrieve the current invert value from a range.
- void **ewl_range_invert_set** (**Ewl_Range** *r, int invert)
Changes the invert property on the range for inverting it's scale.
- double **ewl_range_maximum_value_get** (**Ewl_Range** *r)
Retrieves the maximum value for the range.
- void **ewl_range_maximum_value_set** (**Ewl_Range** *r, double maxv)

Set the maximum value possible for a range.

- `double ewl_range_minimum_value_get (Ewl_Range *r)`
Retrieves the minimum value for the range.
- `void ewl_range_minimum_value_set (Ewl_Range *r, double minv)`
Set the minimum value possible for a range.
- `double ewl_range_step_get (Ewl_Range *r)`
Retrieve the step size of the range.
- `void ewl_range_step_set (Ewl_Range *r, double step)`
Set the steps between increments.
- `double ewl_range_value_get (Ewl_Range *r)`
Retrieve the current value of the range.
- `void ewl_range_value_set (Ewl_Range *r, double v)`
Set the value of pointer of the range.

6.58.2 Define Documentation

6.58.2.1 `#define EWL_RANGE(range) ((Ewl_Range *) range)`

Typecasts a pointer to an `Ewl_Range` (p. 727) pointer.

6.58.2.2 `#define EWL_RANGE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_RANGE_TYPE))`

Returns TRUE if the widget is an `Ewl_Range` (p. 727), FALSE otherwise

6.58.2.3 `#define EWL_RANGE_TYPE "range"`

The type name for the `Ewl_Range` (p. 727) widget

6.58.3 Typedef Documentation

6.58.3.1 `typedef struct Ewl_Range Ewl_Range`

The `Ewl_Range` (p. 727) provides a means to select a value from a range using a draggable button.

6.58.4 Function Documentation

6.58.4.1 `void ewl_range_decrease (Ewl_Range * r)`

Decrease the value of a range by it's step size.

Parameters:

r ,: the range to decrease

Returns:

Returns no value.

Decreases the value of the range r by one increment of it's step size.

6.58.4.2 void ewl_range_increase (Ewl_Range * r)

Increase the value of a range by it's step size.

Parameters:

r ,: the range to increase

Returns:

Returns no value.

Increases the value of the range r by one increment of it's step size.

6.58.4.3 int ewl_range_init (Ewl_Range * r)

Initialize the range to some sane starting values.

Parameters:

r ,: the range to be initialized

Returns:

Returns no value.

Note, there is no `ewl_range_new()`, because range only serves as a common API for seeker, spinner and progressbar

6.58.4.4 int ewl_range_invert_get (Ewl_Range * r)

Retrieve the current invert value from a range.

Parameters:

r ,: the range to retrieve invert property value

Returns:

Returns the current value of the invert property in the range.

6.58.4.5 void ewl_range_invert_set (Ewl_Range * *r*, int *invert*)

Changes the invert property on the range for inverting it's scale.

Parameters:

r,: the range to set invert property

invert,: the new value for the ranges invert property

Returns:

Returns no value.

6.58.4.6 double ewl_range_maximum_value_get (Ewl_Range * *r*)

Retrieves the maximum value for the range.

Parameters:

r,: the range to retrieve maximum value

Returns:

Returns the currently set maximum value for the specified range.

6.58.4.7 void ewl_range_maximum_value_set (Ewl_Range * *r*, double *maxv*)

Set the maximum value possible for a range.

Parameters:

r,: the range to change the maximum possible value

maxv,: the new maximum possible value for *r*

Returns:

Returns no value.

Sets the smallest value that *r* can obtain to *maxv*.

6.58.4.8 double ewl_range_minimum_value_get (Ewl_Range * *r*)

Retrieves the minimum value for the range.

Parameters:

r,: the range to retrieve minimum value

Returns:

Returns the currently set minimum value for the specified range.

6.58.4.9 void ewl_range_minimum_value_set (Ewl_Range * *r*, double *minv*)

Set the minimum value possible for a range.

Parameters:

r,: the range to change the minimum possible value

minv,: the new minimum possible value for *r*

Returns:

Returns no value.

Sets the smallest value that *r* can obtain to *minv*.

6.58.4.10 double ewl_range_step_get (Ewl_Range * *r*)

Retrieve the step size of the range.

Parameters:

r,: the range to retrieve step size

Returns:

Returns the step size of the range *r*.

6.58.4.11 void ewl_range_step_set (Ewl_Range * *r*, double *step*)

Set the steps between increments.

Parameters:

r,: the range to change step

step,: the new step value for the range

Returns:

Returns no value.

Changes the amount that each increment or decrement changes the value of the range *r*.

6.58.4.12 double ewl_range_value_get (Ewl_Range * *r*)

Retrieve the current value of the range.

Parameters:

r,: the ranges to retrieve the value

Returns:

Returns 0 on failure, the value of the range on success.

6.58.4.13 `void ewl_range_value_set (Ewl_Range * r, double v)`

Set the value of pointer of the range.

Parameters:

r,: the range whose value will be changed

v,: the new value of the range, checked against the valid range

Returns:

Returns no value.

6.59 Ewl_Row: The Row widget

6.59.1 Detailed Description

Defines a widget to display a row

Remarks:

Inherits from **Ewl_Container** (p. 618).

Data Structures

- struct **Ewl_Row**

*Inherits from **Ewl_Container** (p. 618) to create a tree row that lays out cells in a left to right fashion.*

Defines

- #define **EWL_ROW**(t) ((**Ewl_Row** *)t)
- #define **EWL_ROW_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_ROW_TYPE**))
- #define **EWL_ROW_TYPE** "row"

Typedefs

- typedef **Ewl_Row** **Ewl_Row**

Functions

- void **ewl_row_cb_child_hide** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_row_cb_child_resize** (**Ewl_Container** *c, **Ewl_Widget** *w, int size, **Ewl_Orientation** o)
- void **ewl_row_cb_child_show** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_row_cb_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_row_cb_header_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_row_cb_header_destroy** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- **Ewl_Widget** * **ewl_row_column_get** (**Ewl_Row** *row, short n)
Retrieve the widget at a specified column.
- void **ewl_row_header_set** (**Ewl_Row** *row, **Ewl_Row** *header)
Set the row header of constraints on cell widths.
- int **ewl_row_init** (**Ewl_Row** *row)
Initialize the row fields of an inheriting object.
- **Ewl_Widget** * **ewl_row_new** (void)
Allocate and initialize a new row.

6.59.2 Define Documentation

6.59.2.1 `#define EWL_ROW(t) ((Ewl_Row *)t)`

Typecasts a pointer to an **Ewl_Row** (p. 729) pointer

6.59.2.2 `#define EWL_ROW_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_ROW_TYPE))`

Returns TRUE if the widget is an **Ewl_Row** (p. 729), FALSE otherwise

6.59.2.3 `#define EWL_ROW_TYPE "row"`

The type name for the **Ewl_Row** (p. 729) widget

6.59.3 Typedef Documentation

6.59.3.1 `typedef struct Ewl_Row Ewl_Row`

The **Ewl_Row** (p. 729) widget

6.59.4 Function Documentation

6.59.4.1 `void ewl_row_cb_child_hide (Ewl_Container * c, Ewl_Widget * w)`

6.59.4.2 `void ewl_row_cb_child_resize (Ewl_Container * c, Ewl_Widget * w, int size, Ewl_Orientation o)`

6.59.4.3 `void ewl_row_cb_child_show (Ewl_Container * c, Ewl_Widget * w)`

6.59.4.4 `void ewl_row_cb_configure (Ewl_Widget * w, void * ev_data, void * user_data)`

6.59.4.5 `void ewl_row_cb_header_configure (Ewl_Widget * w, void * ev_data, void * user_data)`

6.59.4.6 `void ewl_row_cb_header_destroy (Ewl_Widget * w, void * ev_data, void * user_data)`

6.59.4.7 `Ewl_Widget* ewl_row_column_get (Ewl_Row * row, short n)`

Retrieve the widget at a specified column.

Parameters:

row,: the row to retrieve a columns widget from

n,: the column containing the desired widget

Returns:

Returns widget located in column *n* in *row* on success.

6.59.4.8 void ewl_row_header_set (Ewl_Row * *row*, Ewl_Row * *header*)

Set the row header of constraints on cell widths.

Parameters:

row,: the row to change the header row

header,: header row for adjusting cell placement

Returns:

Returns no value.

Changes the row that cell widths and placements will be based on to *header*.

6.59.4.9 int ewl_row_init (Ewl_Row * *row*)

Initialize the row fields of an inheriting object.

Parameters:

row,: the row object to initialize

Returns:

Returns TRUE on success, FALSE on failure.

The fields of the *row* object are initialized to their defaults.

6.59.4.10 Ewl_Widget* ewl_row_new (void)

Allocate and initialize a new row.

Returns:

Returns a newly allocated row on success, NULL on failure.

6.60 Ewl_Scrollbar: A Simple Scrollbar Widget

6.60.1 Detailed Description

Defines an **Ewl_Scrollbar** (p. 730) for using when scrolling values are needed.

Remarks:

Inherits from **Ewl_Box** (p. 586).

Data Structures

- struct **Ewl_Scrollbar**

*Inherits from **Ewl_Box** (p. 586) and extends to provide a scrollbar widget.*

Defines

- #define **EWL_SCROLLBAR**(scrollbar) ((**Ewl_Scrollbar** *) scrollbar)
- #define **EWL_SCROLLBAR_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SCROLLBAR_TYPE))
- #define **EWL_SCROLLBAR_TYPE** "scrollbar"

Typedefs

- typedef **Ewl_Scrollbar** **Ewl_Scrollbar**

Functions

- **Ewl_Widget** * **ewl_hscrollbar_new** (void)
Allocate and initialize a new horizontal scrollbar widget.
- void **ewl_scrollbar_cb_scroll_start** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_scrollbar_cb_scroll_stop** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- int **ewl_scrollbar_init** (**Ewl_Scrollbar** *s)
Initialize a scrollbar to default values.
- char **ewl_scrollbar_inverse_scroll_get** (**Ewl_Scrollbar** *s)
Checks the inverse scrolling flag on a scrollbar.
- void **ewl_scrollbar_inverse_scroll_set** (**Ewl_Scrollbar** *s, char v)
Sets the inverse scrolling flag on a scrollbar.
- **Ewl_Widget** * **ewl_scrollbar_new** (void)
Allocate and initialize a new scrollbar widget.
- **Ewl_Orientation** **ewl_scrollbar_orientation_get** (**Ewl_Scrollbar** *s)

Change the orientation of a scrollbar.

- void **ewl_scrollbar_orientation_set** (**Ewl_Scrollbar** *s, **Ewl_Orientation** orientation)

Change the orientation of a scrollbar.

- double **ewl_scrollbar_step_get** (**Ewl_Scrollbar** *s)

Get the current step size of the scrollbar.

- void **ewl_scrollbar_step_set** (**Ewl_Scrollbar** *s, double v)

Change the step size of a scrollbar.

- double **ewl_scrollbar_value_get** (**Ewl_Scrollbar** *s)

Get the current value of the dragbar.

- void **ewl_scrollbar_value_set** (**Ewl_Scrollbar** *s, double v)

Set the current value of the dragbar.

- **Ewl_Widget** * **ewl_vscrollbar_new** (void)

Allocate and initialize a new vertical scrollbar widget.

6.60.2 Define Documentation

6.60.2.1 #define EWL_SCROLLBAR(scrollbar) ((Ewl_Scrollbar *) scrollbar)

Typecasts a pointer to an **Ewl_Scrollbar** (p. 730) pointer.

6.60.2.2 #define EWL_SCROLLBAR_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SCROLLBAR_TYPE))

Returns TRUE if the widget is an **Ewl_Scrollbar** (p. 730), FALSE otherwise

6.60.2.3 #define EWL_SCROLLBAR_TYPE "scrollbar"

The type name for the **Ewl_Scrollbar** (p. 730) widget

6.60.3 Typedef Documentation

6.60.3.1 typedef struct Ewl_Scrollbar Ewl_Scrollbar

Provides a seeker with increment and decrement buttons arranged in a specified order.

6.60.4 Function Documentation

6.60.4.1 Ewl_Widget* ewl_hscrollbar_new (void)

Allocate and initialize a new horizontal scrollbar widget.

Returns:

Returns NULL on failure, or a pointer to a new scrollbar on success.

6.60.4.2 `void ewl_scrollbar_cb_scroll_start (Ewl_Widget * w, void * ev_data, void * user_data)`

6.60.4.3 `void ewl_scrollbar_cb_scroll_stop (Ewl_Widget * w, void * ev_data, void * user_data)`

6.60.4.4 `int ewl_scrollbar_init (Ewl_Scrollbar * s)`

Initialize a scrollbar to default values.

Parameters:

s,: the scrollbar to initialize

Returns:

Returns no value.

6.60.4.5 `char ewl_scrollbar_inverse_scroll_get (Ewl_Scrollbar * s)`

Checks the inverse scrolling flag on a scrollbar.

Parameters:

s,: the scrollbar to get inverted scrolling value

Returns:

Returns the current value for inverted scrolling.

6.60.4.6 `void ewl_scrollbar_inverse_scroll_set (Ewl_Scrollbar * s, char i)`

Sets the inverse scrolling flag on a scrollbar.

Parameters:

s,: the scrollbar to set inverted scrolling value

i,: value to set for inverse scrolling

Returns:

Returns no value.

6.60.4.7 `Ewl_Widget* ewl_scrollbar_new (void)`

Allocate and initialize a new scrollbar widget.

Returns:

Returns NULL on failure, or a pointer to a new scrollbar on success.

6.60.4.8 Ewl_Orientation ewl_scrollbar_orientation_get (Ewl_Scrollbar * *s*)

Change the orientation of a scrollbar.

Parameters:

s,: the scrollbar to change orientation

Returns:

Returns no value.

6.60.4.9 void ewl_scrollbar_orientation_set (Ewl_Scrollbar * *s*, Ewl_Orientation *o*)

Change the orientation of a scrollbar.

Parameters:

s,: the scrollbar to change orientation

o,: the new orientation to use on the scrollbar

Returns:

Returns no value.

6.60.4.10 double ewl_scrollbar_step_get (Ewl_Scrollbar * *s*)

Get the current step size of the scrollbar.

Parameters:

s,: the scrollbar to retrieve step size

Returns:

Returns the current step size of the scrollbar.

6.60.4.11 void ewl_scrollbar_step_set (Ewl_Scrollbar * *s*, double *v*)

Change the step size of a scrollbar.

Parameters:

s,: the scrollbar to change step size

v,: the new step size of the scrollbar

Returns:

Returns no value.

Changes the step size of the scrollbar *s* to *v*.

6.60.4.12 double ewl_scrollbar_value_get (Ewl_Scrollbar * *s*)

Get the current value of the dragbar.

Parameters:

s,: the scrollbar to get the current value

Returns:

Returns the current value of the scrollbar *s*.

6.60.4.13 void ewl_scrollbar_value_set (Ewl_Scrollbar * *s*, double *v*)

Set the current value of the dragbar.

Parameters:

s,: the scrollbar to set the current value

v,: the new value for the scrollbar

Returns:

Returns no value.

Sets the current value of the scrollbar *s*.

6.60.4.14 Ewl_Widget* ewl_vscrollbar_new (void)

Allocate and initialize a new vertical scrollbar widget.

Returns:

Returns NULL on failure, or a pointer to a new scrollbar on success.

6.61 Ewl_Scrollpane: A Scrollable Viewing Area

6.61.1 Detailed Description

Provides a scrollable area for viewing large sets of widgets in a smaller viewable region.

Remarks:

Inherits from **Ewl_Container** (p. 618).

Data Structures

- struct **Ewl_Scrollpane**

*Inherits from **Ewl_Container** (p. 618) and extends it to enclose a single widget in vertical and horizontal scrollbars for panning around displaying different regions of the enclosed widget.*

Defines

- #define **EWL_SCROLLPANE**(scrollpane) ((**Ewl_Scrollpane** *) scrollpane)
- #define **EWL_SCROLLPANE_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SCROLLPANE_TYPE))
- #define **EWL_SCROLLPANE_TYPE** "scrollpane"

Typedefs

- typedef **Ewl_Scrollpane** **Ewl_Scrollpane**

Functions

- void **ewl_scrollpane_cb_child_resize** (**Ewl_Container** *parent, **Ewl_Widget** *child)
- void **ewl_scrollpane_cb_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_scrollpane_cb_focus_jump** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_scrollpane_cb_hscroll** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_scrollpane_cb_vscroll** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_scrollpane_cb_wheel_scroll** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- **Ewl_Scrollpane_Flags** **ewl_scrollpane_hscrollbar_flag_get** (**Ewl_Scrollpane** *s)
Get flags for horizontal scrollbar.
- void **ewl_scrollpane_hscrollbar_flag_set** (**Ewl_Scrollpane** *s, **Ewl_Scrollpane_Flags** f)
Set flags for horizontal scrollbar.
- double **ewl_scrollpane_hscrollbar_step_get** (**Ewl_Scrollpane** *s)
Retrives the value of the stepping of the vertical scrollbar in s.

- **double ewl_scrollpane_hscrollbar_value_get** (Ewl_Scrollpane *s)
Retrieves the value of the horizontal scrollbar in s.
- **void ewl_scrollpane_hscrollbar_value_set** (Ewl_Scrollpane *s, double val)
Set the value of the horizontal scrollbar in s to val.
- **int ewl_scrollpane_init** (Ewl_Scrollpane *s)
Initialize the fields of a scrollpane.
- **Ewl_Widget * ewl_scrollpane_new** (void)
Create a new scrollpane.
- **Ewl_Scrollpane_Flags ewl_scrollpane_vscrollbar_flag_get** (Ewl_Scrollpane *s)
Get flags for vertical scrollbar.
- **void ewl_scrollpane_vscrollbar_flag_set** (Ewl_Scrollpane *s, Ewl_Scrollpane_Flags f)
Set flags for vertical scrollbar.
- **double ewl_scrollpane_vscrollbar_step_get** (Ewl_Scrollpane *s)
Retrives the value of the stepping of the vertical scrollbar in s.
- **double ewl_scrollpane_vscrollbar_value_get** (Ewl_Scrollpane *s)
Retrieves the value of the vertical scrollbar in s.
- **void ewl_scrollpane_vscrollbar_value_set** (Ewl_Scrollpane *s, double val)
Set the value of the vertical scrollbar in s to val.

6.61.2 Define Documentation

6.61.2.1 `#define EWL_SCROLLPANE(scrollpane) ((Ewl_Scrollpane *) scrollpane)`

Typecasts a pointer to an **Ewl_Scrollpane** (p. 732) pointer.

6.61.2.2 `#define EWL_SCROLLPANE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SCROLLPANE_TYPE))`

Returns TRUE if the widget is an **Ewl_Scrollpane** (p. 732), FALSE otherwise

6.61.2.3 `#define EWL_SCROLLPANE_TYPE "scrollpane"`

The type name for the **Ewl_Scrollpane** (p. 732) widget

6.61.3 Typedef Documentation

6.61.3.1 typedef struct Ewl_Scrollpane Ewl_Scrollpane

The scrollpane provides a way to pan around large collections of images.

6.61.4 Function Documentation

6.61.4.1 void ewl_scrollpane_cb_child_resize (Ewl_Container * *parent*,
Ewl_Widget * *child*)

6.61.4.2 void ewl_scrollpane_cb_configure (Ewl_Widget * *w*, void * *ev_data*,
void * *user_data*)

6.61.4.3 void ewl_scrollpane_cb_focus_jump (Ewl_Widget * *w*, void * *ev_data*,
void * *user_data*)

6.61.4.4 void ewl_scrollpane_cb_hscroll (Ewl_Widget * *w*, void * *ev_data*, void *
user_data)

6.61.4.5 void ewl_scrollpane_cb_vscroll (Ewl_Widget * *w*, void * *ev_data*, void *
user_data)

6.61.4.6 void ewl_scrollpane_cb_wheel_scroll (Ewl_Widget * *w*, void * *ev_data*,
void * *user_data*)

6.61.4.7 Ewl_Scrollpane_Flags ewl_scrollpane_hscrollbar_flag_get
(Ewl_Scrollpane * *s*)

Get flags for horizontal scrollbar.

Parameters:

s,: the scrollpane that contains the scrollbar to retrieve

Returns:

Returns the flags of the horizontal scrollbar, 0 on failure.

6.61.4.8 void ewl_scrollpane_hscrollbar_flag_set (Ewl_Scrollpane * *s*,
Ewl_Scrollpane_Flags *f*)

Set flags for horizontal scrollbar.

Parameters:

s,: the scrollpane that contains the scrollbar to change

f,: the flags to set on the horizontal scrollbar in *s*

Returns:

Returns no value.

The scrollbar flags for the horizontal scrollbar are set to *f*.

6.61.4.9 double ewl_scrollpane_hscrollbar_step_get (Ewl_Scrollpane * *s*)

Retrives the value of the stepping of the vertical scrollbar in *s*.

Parameters:

s,: the scrollpane to retrieve its vertical scrollbar stepping

Returns:

Returns the value of the stepping of the vertical scrollbar in *s* on success.

6.61.4.10 double ewl_scrollpane_hscrollbar_value_get (Ewl_Scrollpane * *s*)

Retrieves the value of the horizontal scrollbar in *s*.

Parameters:

s,: the scrollpane to retrieve it's horizontal scrollbar value

Returns:

Returns the value of the horizontal scrollbar in *s* on success.

6.61.4.11 void ewl_scrollpane_hscrollbar_value_set (Ewl_Scrollpane * *s*, double *val*)

Set the value of the horizontal scrollbar in *s* to *val*.

Parameters:

s,: the scrollpane to set the horizontal scrollbar value

val,: the value to set the scrollbar too

Returns:

Returns nothing

6.61.4.12 int ewl_scrollpane_init (Ewl_Scrollpane * *s*)

Initialize the fields of a scrollpane.

Parameters:

s,: the scrollpane to initialize

Returns:

Returns no value.

Sets up default callbacks and field values for the scrollpane *s*.

6.61.4.13 Ewl_Widget* ewl_scrollpane_new (void)

Create a new scrollpane.

Returns:

Returns a new scrollpane on success, NULL on failure.

6.61.4.14 Ewl_Scrollpane_Flags ewl_scrollpane_vscrollbar_flag_get (Ewl_Scrollpane * *s*)

Get flags for vertical scrollbar.

Parameters:

s,: the scrollpane that contains the scrollbar to retrieve

Returns:

Returns the flags of the vertical scrollbar on success, 0 on failure.

6.61.4.15 void ewl_scrollpane_vscrollbar_flag_set (Ewl_Scrollpane * *s*, Ewl_Scrollpane_Flags *f*)

Set flags for vertical scrollbar.

Parameters:

s,: the scrollpane that contains the scrollbar to change

f,: the flags to set on the vertical scrollbar in *s*

Returns:

Returns no value.

The scrollbar flags for the vertical scrollbar are set to *f*.

6.61.4.16 double ewl_scrollpane_vscrollbar_step_get (Ewl_Scrollpane * *s*)

Retrives the value of the stepping of the vertical scrollbar in *s*.

Parameters:

s,: the scrollpane to retrieve its vertical scrollbar stepping

Returns:

Returns the value of the stepping of the vertical scrollbar in *s* on success.

6.61.4.17 `double ewl_scrollpane_vscrollbar_value_get (Ewl_Scrollpane * s)`

Retrieves the value of the vertical scrollbar in *s*.

Parameters:

s,: the scrollpane to retrieve it's vertical scrollbar value

Returns:

Returns the value of the vertical scrollbar in *s* on success.

6.61.4.18 `void ewl_scrollpane_vscrollbar_value_set (Ewl_Scrollpane * s, double val)`

Set the value of the vertical scrollbar in *s* to *val*.

Parameters:

s,: the scrollpane to set the vertical scrollbar value

val,: the value to set the scrollbar too

Returns:

Returns nothing

6.62 Ewl_Seeker: A Value Selector from a Range

6.62.1 Detailed Description

Defines an **Ewl_Widget** (p. 785) with a draggable button enclosed, used to select a value from a range.

Remarks:

Inherits from **Ewl_Range** (p. 727).

Data Structures

- struct **Ewl_Seeker**

*Inherits from **Ewl_Range** (p. 727) and extends to provide a seeker widget.*

Defines

- #define **EWL_SEEKER**(seeker) ((**Ewl_Seeker** *) seeker)
- #define **EWL_SEEKER_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_SEEKER_TYPE**))
- #define **EWL_SEEKER_TYPE** "seeker"

Typedefs

- typedef **Ewl_Seeker** **Ewl_Seeker**

Functions

- **Ewl_Widget** * **ewl_hseeker_new** (void)
Allocate and initialize a new seeker with horizontal orientation.
- int **ewl_seeker_autohide_get** (**Ewl_Seeker** *s)
Retrieves the current autohide setting on a seeker.
- void **ewl_seeker_autohide_set** (**Ewl_Seeker** *s, int v)
Changes the autohide setting on the seeker to v.
- void **ewl_seeker_cb_button_mouse_down** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_seeker_cb_button_mouse_up** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_seeker_cb_child_show** (**Ewl_Container** *p, **Ewl_Widget** *w)
- void **ewl_seeker_cb_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_seeker_cb_key_down** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_seeker_cb_mouse_down** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_seeker_cb_mouse_move** (**Ewl_Widget** *w, void *ev_data, void *user_data)

- void **ewl_seeker_cb_mouse_up** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- int **ewl_seeker_init** (**Ewl_Seeker** *s)
Initialize the seeker to some sane starting values.
- **Ewl_Widget** * **ewl_seeker_new** (void)
Allocate and initialize a new seeker with default orientation.
- **Ewl_Orientation** **ewl_seeker_orientation_get** (**Ewl_Seeker** *s)
This will retrieve the current orientation set on the seeker.
- void **ewl_seeker_orientation_set** (**Ewl_Seeker** *s, **Ewl_Orientation** o)
Changes the orientation of the given seeker.
- **Ewl_Widget** * **ewl_vseeker_new** (void)
Allocate and initialize a new seeker with vertical orientation.

6.62.2 Define Documentation

6.62.2.1 #define **EWL_SEEKER**(seeker) ((**Ewl_Seeker** *) seeker)

Typecasts a pointer to an **Ewl_Seeker** (p. 734) pointer.

6.62.2.2 #define **EWL_SEEKER_IS**(w) (**ewl_widget_type_is**(**EWL_WIDGET**(w), **EWL_SEEKER_TYPE**))

Returns TRUE if the widget is an **Ewl_Seeker** (p. 734), FALSE otherwise

6.62.2.3 #define **EWL_SEEKER_TYPE** "seeker"

The type name for the **Ewl_Seeker** (p. 734) widget

6.62.3 Typedef Documentation

6.62.3.1 typedef struct **Ewl_Seeker** **Ewl_Seeker**

The **Ewl_Seeker** (p. 734) provides a means to select a value from a range using a draggable button.

6.62.4 Function Documentation

6.62.4.1 **Ewl_Widget*** **ewl_hseeker_new** (void)

Allocate and initialize a new seeker with horizontal orientation.

Returns:

Returns NULL on failure, or a pointer to the new seeker on success.

6.62.4.2 int ewl_seeker_autohide_get (Ewl_Seeker * *s*)

Retrieves the current autohide setting on a seeker.

Parameters:

s,: the seeker to retrieve autohide value

Returns:

Returns TRUE if autohide set, otherwise FALSE.

6.62.4.3 void ewl_seeker_autohide_set (Ewl_Seeker * *s*, int *v*)

Changes the autohide setting on the seeker to *v*.

Parameters:

s,: the seeker to change autohide

v,: the new boolean value for autohiding

Returns:

Returns no value.

Alter the autohide boolean of the seeker *s* to value *v*. If *v* is TRUE, the seeker will be hidden whenever the button is the full size of the seeker.

6.62.4.4 void ewl_seeker_cb_button_mouse_down (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)**6.62.4.5 void ewl_seeker_cb_button_mouse_up (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)****6.62.4.6 void ewl_seeker_cb_child_show (Ewl_Container * *p*, Ewl_Widget * *w*)****6.62.4.7 void ewl_seeker_cb_configure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)****6.62.4.8 void ewl_seeker_cb_key_down (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)****6.62.4.9 void ewl_seeker_cb_mouse_down (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)****6.62.4.10 void ewl_seeker_cb_mouse_move (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)****6.62.4.11 void ewl_seeker_cb_mouse_up (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)****6.62.4.12 int ewl_seeker_init (Ewl_Seeker * *s*)**

Initialize the seeker to some sane starting values.

Parameters:

s,: the seeker to be initialized

Returns:

Returns no value.

Initializes the seeker *s* to the orientation *orientation* to default values and callbacks.

6.62.4.13 Ewl_Widget* ewl_seeker_new (void)

Allocate and initialize a new seeker with default orientation.

Returns:

Returns NULL on failure, or a pointer to the new seeker on success.

6.62.4.14 Ewl_Orientation ewl_seeker_orientation_get (Ewl_Seeker * s)

This will retrieve the current orientation set on the seeker.

Parameters:

s,: The seeker to get the orientation from

Returns:

Returns the orientation set on the seeker

6.62.4.15 void ewl_seeker_orientation_set (Ewl_Seeker * s, Ewl_Orientation o)

Changes the orientation of the given seeker.

Parameters:

s,: the seeker to change orientation

o,: the new orientation for the seeker

Returns:

Returns no value.

6.62.4.16 Ewl_Widget* ewl_vseeker_new (void)

Allocate and initialize a new seeker with vertical orientation.

Returns:

Returns NULL on failure, or a pointer to the new seeker on success.

6.63 Ewl_Separator: A Visual Separator Between Widgets

6.63.1 Detailed Description

Defines the **Ewl_Separator** (p. 740) class used for drawing lines between widgets when visual separation is needed.

Remarks:

Inherits from **Ewl_Widget** (p. 785).

Data Structures

- struct **Ewl_Separator**

*Inherits from **Ewl_Widget** (p. 785) to allow drawing a horizontal or vertical line between widgets, depending on the orientation the **Ewl_Separator** (p. 740).*

Defines

- #define **EWL_SEPARATOR**(separator) ((**Ewl_Separator** *) separator)
- #define **EWL_SEPARATOR_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_SEPARATOR_TYPE**))
- #define **EWL_SEPARATOR_TYPE** "separator"

Typedefs

- typedef **Ewl_Separator** **Ewl_Separator**

Functions

- **Ewl_Widget** * **ewl_hseparator_new** (void)
Allocate a new separator widget with horizontal orientation.
- int **ewl_separator_init** (**Ewl_Separator** *s)
Initialize the separator and inherited fields.
- **Ewl_Widget** * **ewl_separator_new** (void)
Allocate a new separator widget with default (horizontal) orientation.
- **Ewl_Orientation** **ewl_separator_orientation_get** (**Ewl_Separator** *s)
This retrieves the current orientation of the separator.
- void **ewl_separator_orientation_set** (**Ewl_Separator** *s, **Ewl_Orientation** o)
Set the orientation of the separator.
- **Ewl_Widget** * **ewl_vseparator_new** (void)
Allocate a new separator widget with vertical orientation.

6.63.2 Define Documentation

6.63.2.1 `#define EWL_SEPARATOR(separator) ((Ewl_Separator *) separator)`

Typecast a pointer to an **Ewl_Separator** (p. 740) pointer.

6.63.2.2 `#define EWL_SEPARATOR_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SEPARATOR_TYPE))`

Returns TRUE if the widget is an **Ewl_Separator** (p. 740), FALSE otherwise

6.63.2.3 `#define EWL_SEPARATOR_TYPE "separator"`

The type name for the **Ewl_Separator** (p. 740) widget

6.63.3 Typedef Documentation

6.63.3.1 `typedef struct Ewl_Separator Ewl_Separator`

Provides an **Ewl_Widget** (p. 785) to simply act as a separator between other **Ewl_Widget**'s.

6.63.4 Function Documentation

6.63.4.1 `Ewl_Widget* ewl_hseparator_new (void)`

Allocate a new separator widget with horizontal orientation.

Returns:

Returns pointer to new separator widget on success, NULL on failure.

6.63.4.2 `int ewl_separator_init (Ewl_Separator * s)`

Initialize the separator and inherited fields.

Parameters:

s,: the separator to initialize

Returns:

Returns TRUE on success, FALSE on failure.

Clears the contents of the separator and stores the default values.

6.63.4.3 `Ewl_Widget* ewl_separator_new (void)`

Allocate a new separator widget with default (horizontal) orientation.

Returns:

Returns pointer to new separator widget on success, NULL on failure.

6.63.4.4 Ewl_Orientation ewl_separator_orientation_get (Ewl_Separator * *s*)

This retrieves the current orientation of the separator.

Parameters:

s,: The separator to get the orientation from

Returns:

Returns the orientation currently set on the separator

6.63.4.5 void ewl_separator_orientation_set (Ewl_Separator * *s*, Ewl_Orientation *o*)

Set the orientation of the separator.

Parameters:

s,: The separator to set the orientation on

o,: The orientation to set on the separator

Returns:

Returns no value

6.63.4.6 Ewl_Widget* ewl_vseparator_new (void)

Allocate a new separator widget with vertical orientation.

Returns:

Returns pointer to new separator widget on success, NULL on failure.

6.64 Ewl_Shadow: A container with a shadow/border

6.64.1 Detailed Description

Defines the **Ewl_Shadow** (p. 741) class used for adding a shadow decoration around a group of widgets.

Remarks:

Inherits from **Ewl_Box** (p. 586).

Data Structures

- struct **Ewl_Shadow**

*Inherits from **Ewl_Box** (p. 586) to allow drawing a border and shadow decoration around widgets.*

Defines

- #define **EWL_SHADOW**(shadow) ((**Ewl_Shadow** *) shadow)
- #define **EWL_SHADOW_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_SHADOW_TYPE**))
- #define **EWL_SHADOW_TYPE** "shadow"

Typedefs

- typedef **Ewl_Shadow** **Ewl_Shadow**

Functions

- int **ewl_shadow_init** (**Ewl_Shadow** *s)
Initialize a shadow container to default values.
- **Ewl_Widget** * **ewl_shadow_new** (void)
Allocate and initialize a new shadow container.

6.64.2 Define Documentation

6.64.2.1 #define **EWL_SHADOW**(shadow) ((**Ewl_Shadow** *) shadow)

Typecast a pointer to an **Ewl_Shadow** (p. 741) pointer.

6.64.2.2 #define **EWL_SHADOW_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_SHADOW_TYPE**))

Returns TRUE if the widget is an **Ewl_Shadow** (p. 741), FALSE otherwise

6.64.2.3 #define EWL_SHADOW_TYPE "shadow"

The type name for the `Ewl_Shadow` (p. 741) widget

6.64.3 Typedef Documentation

6.64.3.1 typedef struct Ewl_Shadow Ewl_Shadow

`Ewl_Shadow` (p. 741) provides a container that has a shadowed border.

6.64.4 Function Documentation

6.64.4.1 int ewl_shadow_init (Ewl_Shadow * s)

Initialize a shadow container to default values.

Parameters:

s,: the shadow container to initialize

Returns:

Returns TRUE on success, FALSE on failure.

6.64.4.2 Ewl_Widget* ewl_shadow_new (void)

Allocate and initialize a new shadow container.

Returns:

Returns a new shadow container on success, NULL on failure.

6.65 Ewl_Spacer: A Widget to Add Space Between Other Widgets

6.65.1 Detailed Description

Defines an **Ewl_Widget** (p. 785) to be used for adding space between widgets in a layout.

Remarks:

Inherits from **Ewl_Widget** (p. 785).

Data Structures

- struct **Ewl_Spacer**

*Inherits from **Ewl_Widget** (p. 785) and provides as a spacer between widgets.*

Defines

- `#define EWL_SPACER(spacer) ((Ewl_Spacer *) spacer)`
- `#define EWL_SPACER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SPACER_TYPE))`
- `#define EWL_SPACER_TYPE "spacer"`

Typedefs

- `typedef Ewl_Spacer Ewl_Spacer`

Functions

- `int ewl_spacer_init (Ewl_Spacer *s)`
Initialize a spacer to starting values.
- `Ewl_Widget * ewl_spacer_new (void)`
Allocate and initialize a new spacer.

6.65.2 Define Documentation

6.65.2.1 `#define EWL_SPACER(spacer) ((Ewl_Spacer *) spacer)`

Typecasts a pointer to an **Ewl_Spacer** (p. 742) pointer.

6.65.2.2 `#define EWL_SPACER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SPACER_TYPE))`

Returns TRUE if the widget is an **Ewl_Spacer** (p. 742), FALSE otherwise

6.65.2.3 `#define EWL_SPACER_TYPE "spacer"`

The type name for the `Ewl_Spacer` (p. 742) widget

6.65.3 Typedef Documentation

6.65.3.1 `typedef struct Ewl_Spacer Ewl_Spacer`

The `Ewl_Spacer` (p. 742) is a simple widget that is used to tweak spacing in an app.

6.65.4 Function Documentation

6.65.4.1 `int ewl_spacer_init (Ewl_Spacer * s)`

Initialize a spacer to starting values.

Parameters:

`s`, the spacer to initialize

Returns:

Returns TRUE on success or FALSE on failure

Initializes a spacer to default values and callbacks.

6.65.4.2 `Ewl_Widget* ewl_spacer_new (void)`

Allocate and initialize a new spacer.

Returns:

Returns NULL on failure, a pointer to a new spacer on success

6.66 Ewl_Spectrum: The colour spectrum widget

6.66.1 Detailed Description

Defines a colour spectrum widget

Remarks:

Inherits from **Ewl_Overlay** (p. 712).

Data Structures

- struct **Ewl_Spectrum**

*Inherits from **Ewl_Overlay** (p. 712) and extends to provide a colour spectrum.*

Defines

- #define **EWL_SPECTRUM**(sp) ((**Ewl_Spectrum** *)sp)
- #define **EWL_SPECTRUM_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_SPECTRUM_TYPE**))
- #define **EWL_SPECTRUM_TYPE** "spectrum"

Typedefs

- typedef **Ewl_Spectrum** **Ewl_Spectrum**

Functions

- void **ewl_spectrum_canvas_cb_reveal** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_spectrum_cb_configure** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_spectrum_cb_mouse_down** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_spectrum_cb_mouse_move** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_spectrum_cb_mouse_up** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_spectrum_hsv_get** (**Ewl_Spectrum** *sp, double *h, double *s, double *v)

Get the HSV value from the spectrum.

- void **ewl_spectrum_hsv_set** (**Ewl_Spectrum** *sp, double h, double s, double v)

Set the HSV values for the spectrum.

- int **ewl_spectrum_init** (**Ewl_Spectrum** *sp)

*Initializes an **Ewl_Spectrum** widget to default values.*

- **Ewl_Color_Mode** **ewl_spectrum_mode_get** (**Ewl_Spectrum** *sp)

Get the mode of the spectrum.

- void **ewl_spectrum_mode_set** (**Ewl_Spectrum** *sp, **Ewl_Color_Mode** mode)

Set the mode of the spectrum.

- **Ewl_Widget * ewl_spectrum_new** (void)

*Creates a new **Ewl_Spectrum** (p. 743) widget.*

- **void ewl_spectrum_rgb_get** (**Ewl_Spectrum** *sp, unsigned int *r, unsigned int *g, unsigned int *b)

Get the RGB values for the spectrum.

- **void ewl_spectrum_rgb_set** (**Ewl_Spectrum** *sp, unsigned int r, unsigned int g, unsigned int b)

Set the RGB values for the spectrum.

- **Ewl_Spectrum_Type ewl_spectrum_type_get** (**Ewl_Spectrum** *sp)

Get the type of the spectrum.

- **void ewl_spectrum_type_set** (**Ewl_Spectrum** *sp, **Ewl_Spectrum_Type** type)

Set the type of the spectrum.

6.66.2 Define Documentation

6.66.2.1 **#define EWL_SPECTRUM(sp) ((Ewl_Spectrum *)sp)**

Typecasts a pointer to an **Ewl_Spectrum** (p. 743) pointer.

6.66.2.2 **#define EWL_SPECTRUM_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SPECTRUM_TYPE))**

Returns TRUE if the widget is an **Ewl_Spectrum** (p. 743), FALSE otherwise

6.66.2.3 **#define EWL_SPECTRUM_TYPE "spectrum"**

The type name for the **Ewl_Spectrum** (p. 743) widget

6.66.3 Typedef Documentation

6.66.3.1 **typedef struct Ewl_Spectrum Ewl_Spectrum**

Provides a colour spectrum.

6.66.4 Function Documentation

6.66.4.1 `void ewl_spectrum_canvas_cb_reveal (Ewl_Widget * w, void * ev, void * data)`

6.66.4.2 `void ewl_spectrum_cb_configure (Ewl_Widget * w, void * ev, void * data)`

6.66.4.3 `void ewl_spectrum_cb_mouse_down (Ewl_Widget * w, void * ev, void * data)`

6.66.4.4 `void ewl_spectrum_cb_mouse_move (Ewl_Widget * w, void * ev, void * data)`

6.66.4.5 `void ewl_spectrum_cb_mouse_up (Ewl_Widget * w, void * ev, void * data)`

6.66.4.6 `void ewl_spectrum_hsv_get (Ewl_Spectrum * sp, double * h, double * s, double * v)`

Get the HSV value from the spectrum.

Parameters:

sp,: The **Ewl_Spectrum** (p. 743) to get the hsv values from

h,: Where to place the hue

s,: Where to place the saturation

v,: Where to place the value

Returns:

Returns no value.

6.66.4.7 `void ewl_spectrum_hsv_set (Ewl_Spectrum * sp, double h, double s, double v)`

Set the HSV values for the spectrum.

Parameters:

sp,: The **Ewl_Spectrum** (p. 743) to set the hsv value into

h,: The hue to set

s,: The saturation to set

v,: The value to set

Returns:

Returns no value

6.66.4.8 int ewl_spectrum_init (Ewl_Spectrum * *sp*)

Initializes an Ewl_Spectrum widget to default values.

Parameters:

sp,: The **Ewl_Spectrum** (p. 743) to init

Returns:

Returns TRUE on success or FALSE on failure

6.66.4.9 Ewl_Color_Mode ewl_spectrum_mode_get (Ewl_Spectrum * *sp*)

Get the mode of the spectrum.

Parameters:

sp,: The **Ewl_Spectrum** (p. 743) to get the mode from

Returns:

Returns the mode of the spectrum

6.66.4.10 void ewl_spectrum_mode_set (Ewl_Spectrum * *sp*, Ewl_Color_Mode *mode*)

Set the mode of the spectrum.

Parameters:

sp,: The **Ewl_Spectrum** (p. 743) to set the mode on

mode,: The mode to set the spectrum too

Returns:

Returns no value.

6.66.4.11 Ewl_Widget* ewl_spectrum_new (void)

Creates a new **Ewl_Spectrum** (p. 743) widget.

Returns:

Returns a new **Ewl_Spectrum** (p. 743) widget or NULL on failure

6.66.4.12 `void ewl_spectrum_rgb_get (Ewl_Spectrum * sp, unsigned int * r, unsigned int * g, unsigned int * b)`

Get the RGB values for the spectrum.

Parameters:

sp,: The **Ewl_Spectrum** (p. 743) to get the rgb values from

r,: Where to store the red value

g,: Where to store the green value

b,: Where to store the blue value

Returns:

Returns no value.

6.66.4.13 `void ewl_spectrum_rgb_set (Ewl_Spectrum * sp, unsigned int r, unsigned int g, unsigned int b)`

Set the RGB values for the spectrum.

Parameters:

sp,: The **Ewl_Spectrum** (p. 743) to set the colour into

r,: The red value to set

g,: The green value to set

b,: The blue value to set

Returns:

Returns no value

6.66.4.14 `Ewl_Spectrum_Type ewl_spectrum_type_get (Ewl_Spectrum * sp)`

Get the type of the spectrum.

Parameters:

sp,: The **Ewl_Spectrum** (p. 743) to get the type from

Returns:

Returns the spectrum type

6.66.4.15 `void ewl_spectrum_type_set (Ewl_Spectrum * sp, Ewl_Spectrum_Type type)`

Set the type of the spectrum.

Parameters:

sp,: The **Ewl_Spectrum** (p. 743) to set the type on

type,: The type to set the spectrum too

Returns:

Returns no value

6.67 Ewl_Spinner: A Numerical Value Entry

6.67.1 Detailed Description

Provides a field for entering numerical values, along with buttons to increment and decrement the value.

Remarks:

Inherits from **Ewl_Range** (p. 727).

Data Structures

- struct **Ewl_Spinner**

*Inherits from **Ewl_Range** (p. 727) and extends to provide a spinner widget.*

Defines

- `#define EWL_SPINNER(spinner) ((Ewl_Spinner *) spinner)`
- `#define EWL_SPINNER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SPINNER_TYPE))`
- `#define EWL_SPINNER_TYPE "spinner"`

Typedefs

- `typedef Ewl_Spinner Ewl_Spinner`

Functions

- `void ewl_spinner_cb_child_resize (Ewl_Container *c, Ewl_Widget *w, int size, Ewl_Orientation o)`
- `void ewl_spinner_cb_child_show (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_spinner_cb_configure (Ewl_Widget *widget, void *ev_data, void *user_data)`
- `void ewl_spinner_cb_decrease_value (Ewl_Widget *widget, void *ev_data, void *user_data)`
- `void ewl_spinner_cb_destroy (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_spinner_cb_focus_out (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_spinner_cb_increase_value (Ewl_Widget *widget, void *ev_data, void *user_data)`
- `void ewl_spinner_cb_key_down (Ewl_Widget *widget, void *ev_data, void *user_data)`
- `void ewl_spinner_cb_realize (Ewl_Widget *widget, void *ev_data, void *user_data)`
- `void ewl_spinner_cb_value_changed (Ewl_Widget *widget, void *ev_data, void *user_data)`
- `void ewl_spinner_cb_value_stop (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_spinner_cb_wheel (Ewl_Widget *w, void *ev_data, void *user_data)`
- `unsigned char ewl_spinner_digits_get (Ewl_Spinner *s)`

This will retrieve the number of digits displayed by the spinner.

- void **ewl_spinner_digits_set** (**Ewl_Spinner** *s, unsigned char digits)

Set the number of digits displayed by a spinner.

- int **ewl_spinner_init** (**Ewl_Spinner** *s)

Initialize a spinner to default values and callbacks.

- **Ewl_Widget** * **ewl_spinner_new** (void)

Allocate a new spinner widget.

6.67.2 Define Documentation

6.67.2.1 #define **EWL_SPINNER(spinner)** ((**Ewl_Spinner** *) spinner)

Typecasts a pointer to an **Ewl_Spinner** (p. 746) pointer.

6.67.2.2 #define **EWL_SPINNER_IS(w)** (**ewl_widget_type_is**(**EWL_WIDGET(w)**, **EWL_SPINNER_TYPE**))

Returns TRUE if the widget is an **Ewl_Spinner** (p. 746), FALSE otherwise

6.67.2.3 #define **EWL_SPINNER_TYPE** "spinner"

The type name for the **Ewl_Spinner** (p. 746) widget

6.67.3 Typedef Documentation

6.67.3.1 typedef struct **Ewl_Spinner** **Ewl_Spinner**

A combination of entry and increment/decrement buttons for adjusting numerical values.

6.67.4 Function Documentation

- 6.67.4.1 `void ewl_spinner_cb_child_resize (Ewl_Container * c, Ewl_Widget * w, int size, Ewl_Orientation o)`
- 6.67.4.2 `void ewl_spinner_cb_child_show (Ewl_Container * c, Ewl_Widget * w)`
- 6.67.4.3 `void ewl_spinner_cb_configure (Ewl_Widget * widget, void * ev_data, void * user_data)`
- 6.67.4.4 `void ewl_spinner_cb_decrease_value (Ewl_Widget * widget, void * ev_data, void * user_data)`
- 6.67.4.5 `void ewl_spinner_cb_destroy (Ewl_Widget * w, void * ev_data, void * user_data)`
- 6.67.4.6 `void ewl_spinner_cb_focus_out (Ewl_Widget * w, void * ev_data, void * user_data)`
- 6.67.4.7 `void ewl_spinner_cb_increase_value (Ewl_Widget * widget, void * ev_data, void * user_data)`
- 6.67.4.8 `void ewl_spinner_cb_key_down (Ewl_Widget * widget, void * ev_data, void * user_data)`
- 6.67.4.9 `void ewl_spinner_cb_realize (Ewl_Widget * widget, void * ev_data, void * user_data)`
- 6.67.4.10 `void ewl_spinner_cb_value_changed (Ewl_Widget * widget, void * ev_data, void * user_data)`
- 6.67.4.11 `void ewl_spinner_cb_value_stop (Ewl_Widget * w, void * ev_data, void * user_data)`
- 6.67.4.12 `void ewl_spinner_cb_wheel (Ewl_Widget * w, void * ev_data, void * user_data)`
- 6.67.4.13 `unsigned char ewl_spinner_digits_get (Ewl_Spinner * s)`

This will retrieve the number of digits displayed by the spinner.

Parameters:

s,: The spinner to get the number of digits displayed

Returns:

Returns the number of digits displayed by the spinner

- 6.67.4.14 `void ewl_spinner_digits_set (Ewl_Spinner * s, unsigned char digits)`

Set the number of digits displayed by a spinner.

Parameters:

s,: the widget to change the number of digits displayed
digits,: the number of digits to display for the spinner *s*

Returns:

Returns no value.

Changes the digits displayed by *s* to *digits*.

6.67.4.15 int ewl_spinner_init (Ewl_Spinner * *s*)

Initialize a spinner to default values and callbacks.

Parameters:

s,: the spinner to initialize

Returns:

Returns TRUE on success or FALSE on failure

Sets the fields and callbacks of the spinner *s* their default values.

6.67.4.16 Ewl_Widget* ewl_spinner_new (void)

Allocate a new spinner widget.

Returns:

Returns a new spinner widget on success, NULL on failure.

6.68 Ewl_Statusbar: A status bar widget

6.68.1 Detailed Description

Defines a class for displaying status bars

Remarks:

Inherits from **Ewl_Box** (p. 586).

Data Structures

- struct **Ewl_Statusbar**

*Inherits from the **Ewl_Box** (p. 586) class and extends it to provide for a simple status bar.*

Defines

- #define **EWL_STATUSBAR**(statusbar) ((**Ewl_Statusbar** *) statusbar)
- #define **EWL_STATUSBAR_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_STATUSBAR_TYPE**))
- #define **EWL_STATUSBAR_TYPE** "statusbar"

Typedefs

- typedef **Ewl_Statusbar** **Ewl_Statusbar**

Functions

- void **ewl_statusbar_active_set** (**Ewl_Statusbar** *sb, **Ewl_Position** pos)
Sets the active statusbar container.
- void **ewl_statusbar_cb_destroy** (**Ewl_Widget** *w, void *ev, void *data)
- int **ewl_statusbar_init** (**Ewl_Statusbar** *sb)
Initialize a statusbar to default values.
- void **ewl_statusbar_left_append** (**Ewl_Statusbar** *sb, **Ewl_Widget** *w)
pack the given widget w into the statusbar sb into right of the left box.
- void **ewl_statusbar_left_hide** (**Ewl_Statusbar** *sb)
hide the box on the left of the status bar
- void **ewl_statusbar_left_prepend** (**Ewl_Statusbar** *sb, **Ewl_Widget** *w)
pack the given widget w into the statusbar sb into the left of the left box
- void **ewl_statusbar_left_show** (**Ewl_Statusbar** *sb)
show the box on the left of the status bar
- **Ewl_Widget** * **ewl_statusbar_new** (void)

Allocate and initialize a new statusbar widget.

- **void ewl_statusbar_pop (Ewl_Statusbar *sb)**
Removes the most recent status message and displays the one before it.
- **void ewl_statusbar_push (Ewl_Statusbar *sb, char *txt)**
Displays the given status message on the statusbar.
- **void ewl_statusbar_right_append (Ewl_Statusbar *sb, Ewl_Widget *w)**
pack the given widget w into the statusbar sb into the right of the right box
- **void ewl_statusbar_right_hide (Ewl_Statusbar *sb)**
hide the box on the right of the status bar
- **void ewl_statusbar_right_prepend (Ewl_Statusbar *sb, Ewl_Widget *w)**
pack the given widget w into the statusbar sb into the left of the right box
- **void ewl_statusbar_right_show (Ewl_Statusbar *sb)**
show the box on the right of the status bar

6.68.2 Define Documentation

6.68.2.1 #define EWL_STATUSBAR(statusbar) ((Ewl_Statusbar *) statusbar)

Typecasts a pointer to a **Ewl_Statusbar** (p. 748) pointer

6.68.2.2 #define EWL_STATUSBAR_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_STATUSBAR_TYPE))

Returns TRUE if the widget is an **Ewl_Statusbar** (p. 748), FALSE otherwise

6.68.2.3 #define EWL_STATUSBAR_TYPE "statusbar"

The type name for the **Ewl_Statusbar** (p. 748) widget

6.68.3 Typedef Documentation

6.68.3.1 typedef struct Ewl_Statusbar Ewl_Statusbar

The **Ewl_Statusbar** (p. 748)

6.68.4 Function Documentation

6.68.4.1 void ewl_statusbar_active_set (Ewl_Statusbar * sb, Ewl_Position pos)

Sets the active statusbar container.

Parameters:

sb,: The status bar to work on

pos,: The side of the statusbar to set active either EWL_POSITION_LEFT or EWL_POSITION_RIGHT

Returns:

Returns no value

6.68.4.2 void ewl_statusbar_cb_destroy (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.68.4.3 int ewl_statusbar_init (Ewl_Statusbar * *sb*)

Initialize a statusbar to default values.

Parameters:

sb,: the statusbar to initialize

Returns:

Returns 1 on success and 0 on failure

6.68.4.4 void ewl_statusbar_left_append (Ewl_Statusbar * *sb*, Ewl_Widget * *w*)

pack the given widget *w* into the statusbar *sb* into right of the left box.

Parameters:

sb,: The statusbar to operate on

w,: The widget to add to the container

Returns:

Returns no value

6.68.4.5 void ewl_statusbar_left_hide (Ewl_Statusbar * *sb*)

hide the box on the left of the status bar

Parameters:

sb,: The statusbar to work on

Returns:

Returns no value

6.68.4.6 void ewl_statusbar_left_prepend (Ewl_Statusbar * *sb*, Ewl_Widget * *w*)

pack the given widget *w* into the statusbar *sb* into the left of the left box

Parameters:

- sb*,: The statusbar to operate on
- w*,: The widget to add to the container

Returns:

Returns no value

6.68.4.7 void ewl_statusbar_left_show (Ewl_Statusbar * *sb*)

show the box on the left of the status bar

Parameters:

- sb*,: The statusbar to work on

Returns:

Returns no value

6.68.4.8 Ewl_Widget* ewl_statusbar_new (void)

Allocate and initialize a new statusbar widget.

Returns:

Returns NULL on failure, or a pointer to a new statusbar on success.

6.68.4.9 void ewl_statusbar_pop (Ewl_Statusbar * *sb*)

Removes the most recent status message and displays the one before it.

Parameters:

- sb*,: the status bar to remove the status mesasge from

Returns:

Returns no value

6.68.4.10 void ewl_statusbar_push (Ewl_Statusbar * *sb*, char * *txt*)

Displays the given status message on the statusbar.

Parameters:

- sb*,: the statusbar to add the message too

txt,: the status message to display

Returns:

Returns no value

6.68.4.11 `void ewl_statusbar_right_append (Ewl_Statusbar * sb, Ewl_Widget * w)`

pack the given widget *w* into the statusbar *sb* into the right of the right box

Parameters:

sb,: The statusbar to operate on

w,: The widget to add to the container

Returns:

Returns no value

6.68.4.12 `void ewl_statusbar_right_hide (Ewl_Statusbar * sb)`

hide the box on the right of the status bar

Parameters:

sb,: The statusbar to work on

Returns:

Returns no value

6.68.4.13 `void ewl_statusbar_right_prepend (Ewl_Statusbar * sb, Ewl_Widget * w)`

pack the given widget *w* into the statusbar *sb* into the left of the right box

Parameters:

sb,: The statusbar to operate on

w,: The widget to add to the container

Returns:

Returns no value

6.68.4.14 void ewl_statusbar_right_show (Ewl_Statusbar * *sb*)

show the box on the right of the status bar

Parameters:

sb,: The statusbar to work on

Returns:

Returns no value

6.69 Ewl_Stock: The Stock

6.69.1 Detailed Description

The stock class is an abstract class to provide the stock feature of buttons and icons in a common place. This class inherits from the **Ewl_Box** (p. 586) to allow for placing any other widget inside the stock.

Remarks:

Inherits from **Ewl_Box** (p. 586).

Data Structures

- struct **Ewl_Stock**
*Inherits from **Ewl_Box** (p. 586) and provides the base for the other stock icons.*
- struct **Ewl_Stock_Funcs**
The function callbacks that an inheriting widget needs to implement.

Defines

- `#define EWL_STOCK(stock) ((Ewl_Stock *) stock)`
- `#define EWL_STOCK_IMAGE_SET(f) ((Ewl_Stock_Image_Set)f)`
- `#define EWL_STOCK_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_STOCK_TYPE))`
- `#define EWL_STOCK_LABEL_SET(f) ((Ewl_Stock_Label_Set)f)`
- `#define EWL_STOCK_TOOLTIP_SET(f) ((Ewl_Stock_Tooltip_Set)f)`
- `#define EWL_STOCK_TYPE "stock"`

Typedefs

- `typedef Ewl_Stock Ewl_Stock`
- `typedef Ewl_Stock_Funcs Ewl_Stock_Funcs`
- `typedef void(*) Ewl_Stock_Image_Set (Ewl_Stock *s, const char *file, const char *key)`
- `typedef void(*) Ewl_Stock_Label_Set (Ewl_Stock *s, const char *txt)`
- `typedef void(*) Ewl_Stock_Tooltip_Set (Ewl_Stock *s, const char *tip)`

Functions

- `void ewl_stock_functions_set (Ewl_Stock *s, Ewl_Stock_Funcs *funcs)`
- `int ewl_stock_init (Ewl_Stock *s)`
Initialize the stock widget and inherited fields.
- `Ewl_Stock_Type ewl_stock_type_get (Ewl_Stock *s)`
Get the stock type in use.
- `void ewl_stock_type_set (Ewl_Stock *s, Ewl_Stock_Type stock)`

Set the type of the stock to use.

6.69.2 Define Documentation

6.69.2.1 `#define EWL_STOCK(stock) ((Ewl_Stock *) stock)`

Typecast a pointer to an **Ewl_Stock** (p. 750) pointer.

6.69.2.2 `#define EWL_STOCK_IMAGE_SET(f) ((Ewl_Stock_Image_Set)f)`

Typecasts a pointer to an image set pointer

6.69.2.3 `#define EWL_STOCK_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_STOCK_TYPE))`

Returns TRUE if the widget is an **Ewl_Stock** (p. 750), FALSE otherwise

6.69.2.4 `#define EWL_STOCK_LABEL_SET(f) ((Ewl_Stock_Label_Set)f)`

Typecasts a pointer to a label set pointer

6.69.2.5 `#define EWL_STOCK_TOOLTIP_SET(f) ((Ewl_Stock_Tooltip_Set)f)`

Typecast to the tooltip set pointer

6.69.2.6 `#define EWL_STOCK_TYPE "stock"`

The type name for the **Ewl_Stock** (p. 750) widget

6.69.3 Typedef Documentation

6.69.3.1 `typedef struct Ewl_Stock Ewl_Stock`

The stock structure.

6.69.3.2 `typedef struct Ewl_Stock_Funcs Ewl_Stock_Funcs`

6.69.3.3 `typedef void(*) Ewl_Stock_Image_Set(Ewl_Stock *s, const char *file, const char *key)`

The image set function definition

6.69.3.4 `typedef void(*) Ewl_Stock_Label_Set(Ewl_Stock *s, const char *txt)`

The label set function definition

6.69.3.5 typedef void(*) Ewl_Stock_Tooltip_Set(Ewl_Stock *s, const char *tip)

The tooltip set function definition

6.69.4 Function Documentation

6.69.4.1 void ewl_stock_functions_set (Ewl_Stock * s, Ewl_Stock_Funcs * *funcs*)

6.69.4.2 int ewl_stock_init (Ewl_Stock * s)

Initialize the stock widget and inherited fields.

Parameters:

s,: the stock widget to initialize

Returns:

Returns TRUE on success, FALSE on failure.

Clears the contents of the stock widget and stores the default values.

6.69.4.3 Ewl_Stock_Type ewl_stock_type_get (Ewl_Stock * s)

Get the stock type in use.

Parameters:

s,: The stock widget to get the stock type from

Returns:

Returns the Ewl_Stock_Type of the stock widget

6.69.4.4 void ewl_stock_type_set (Ewl_Stock * s, Ewl_Stock_Type *stock*)

Set the type of the stock to use.

Parameters:

s,: The stock to set the stock type on

stock,: The Ewl_Stock_Type to set on the stock widget

Returns:

Returns no value.

6.70 Ewl_Table: The Table Layout Container.

6.70.1 Detailed Description

Defines the **Ewl_Box** (p. 586) class used for laying out Ewl_Widget's in a horizontal or vertical line.

Remarks:

Inherits from **Ewl_Container** (p. 618).

Data Structures

- struct **Ewl_Table**

*Inherits from **Ewl_Container** (p. 618) and extends to provide a table layout.*

Defines

- #define **EWL_TABLE**(table) ((**Ewl_Table** *)table)
- #define **EWL_TABLE_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TABLE_TYPE))
- #define **EWL_TABLE_TYPE** "table"

Typedefs

- typedef **Ewl_Table** **Ewl_Table**

Functions

- void **ewl_table_add** (**Ewl_Table** *table, **Ewl_Widget** *w, int start_col, int end_col, int start_row, int end_row)
Add a child widget to the table.
- void **ewl_table_cb_child_select** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_table_cb_child_show** (**Ewl_Container** *p, **Ewl_Widget** *c)
- void **ewl_table_cb_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_table_col_row_get** (**Ewl_Table** *table, **Ewl_Cell** *cell, int *start_col, int *end_col, int *start_row, int *end_row)
Get the column and row of a widget.
- void **ewl_table_column_w_get** (**Ewl_Table** *table, int col, int *width)
Get the width of a table column.
- void **ewl_table_column_w_set** (**Ewl_Table** *table, int col, int width)
Set the width of a table column.
- **Ecore_List** * **ewl_table_find** (**Ewl_Table** *table, int start_col, int end_col, int start_row, int end_row)

Get a list of the widgets in the specified col/row.

- unsigned int **ewl_table_hhomogeneous_get** (**Ewl_Table** *table)
Retrieves the horizontal homogeneous flag.
- void **ewl_table_hhomogeneous_set** (**Ewl_Table** *table, unsigned int h)
Change the horizontal homogeneous layout of the box.
- void **ewl_table_homogeneous_set** (**Ewl_Table** *table, unsigned int h)
Change the homogeneous layout of the box.
- int **ewl_table_init** (**Ewl_Table** *t, int cols, int rows, char **col_headers)
Initialize table to starting values.
- **Ewl_Widget** * **ewl_table_new** (int cols, int rows, char **col_headers)
Create a new table.
- void **ewl_table_reset** (**Ewl_Table** *t, int cols, int rows, char **c_headers)
Clear the table and set new geometry.
- void **ewl_table_row_h_get** (**Ewl_Table** *table, int row, int *height)
Get the height of a table row.
- void **ewl_table_row_h_set** (**Ewl_Table** *table, int row, int height)
Set the height of a table row.
- char * **ewl_table_selected_get** (**Ewl_Table** *table)
Get the text in the current selected box.
- unsigned int **ewl_table_vhomogeneous_get** (**Ewl_Table** *table)
Retrieves the vertical homogeneous flag.
- void **ewl_table_vhomogeneous_set** (**Ewl_Table** *table, unsigned int h)
Change the vertical homogeneous layout of the box.

6.70.2 Define Documentation

6.70.2.1 #define EWL_TABLE(table) ((Ewl_Table *)table)

Typecase a pointer to an **Ewl_Table** (p. 753) pointer.

6.70.2.2 #define EWL_TABLE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TABLE_TYPE))

Returns TRUE if the widget is an **Ewl_Table** (p. 753), FALSE otherwise

6.70.2.3 #define EWL_TABLE_TYPE "table"

The type name for the **Ewl_Table** (p. 753) widget

6.70.3 Typedef Documentation

6.70.3.1 typedef struct Ewl_Table Ewl_Table

The table widget is an **Ewl_Container** (p.618) used to lay out widgets in a grid like pattern with headers and alignment.

6.70.4 Function Documentation

6.70.4.1 void ewl_table_add (Ewl_Table * *table*, Ewl_Widget * *w*, int *start_col*, int *end_col*, int *start_row*, int *end_row*)

Add a child widget to the table.

Parameters:

table,: the table
w,: the cell to add
start_col,: the start column
end_col,: the end column
start_row,: the start row
end_row,: the end row

Returns:

Returns no value.

6.70.4.2 void ewl_table_cb_child_select (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.70.4.3 void ewl_table_cb_child_show (Ewl_Container * *p*, Ewl_Widget * *c*)

6.70.4.4 void ewl_table_cb_configure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.70.4.5 void ewl_table_col_row_get (Ewl_Table * *t*, Ewl_Cell * *cell*, int * *start_col*, int * *end_col*, int * *start_row*, int * *end_row*)

Get the column and row of a widget.

Parameters:

t,: the table
cell,: the cell to add
start_col,: integer pointer to store the start column
end_col,: integer pointer to store the end column in
start_row,: integer pointer to store the start row in
end_row,: integer pointer to store the end row in

Returns:

Returns nothing

6.70.4.6 void ewl_table_column_w_get (Ewl_Table * *table*, int *col*, int * *width*)

Get the width of a table column.

Parameters:

table,: the table

col,: the column

width,: integer pointer to store the width in

Returns:

Returns no value.

6.70.4.7 void ewl_table_column_w_set (Ewl_Table * *table*, int *col*, int *width*)

Set the width of a table column.

Parameters:

table,: the table

col,: the column

width,: the new width

Returns:

Returns no value.

6.70.4.8 Ecore_List* ewl_table_find (Ewl_Table * *t*, int *start_col*, int *end_col*, int *start_row*, int *end_row*)

Get a list of the widgets in the specified col/row.

Parameters:

t,: the table

start_col,: the start column

end_col,: the end column

start_row,: the start row

end_row,: the end row

Returns:

Returns Ecore_List of widgets found in the specified col/row area.

6.70.4.9 unsigned int ewl_table_hhomogeneous_get (Ewl_Table * *table*)

Retrieves the horizontal homogeneous flag.

Parameters:

table,: the table to get the homogeneous layout

Returns:

The horizontal homogeneous flag

6.70.4.10 void ewl_table_hhomogeneous_set (Ewl_Table * *table*, unsigned int *h*)

Change the horizontal homogeneous layout of the box.

Parameters:

table,: the table to change horizontal homogeneous layout

h,: the boolean value to change the horizontal layout mode to

Returns:

Returns no value.

Grids use non-homogeneous layout by default, this can be used to change that for horizontal orientation, i.e. all columns can have the same width.

6.70.4.11 void ewl_table_homogeneous_set (Ewl_Table * *table*, unsigned int *h*)

Change the homogeneous layout of the box.

Parameters:

table,: the table to change homogeneous layout

h,: the boolean value to change the layout mode to

Returns:

Returns no value.

Grids use non-homogeneous layout by default, this can be used to change that.

**6.70.4.12 int ewl_table_init (Ewl_Table * *t*, int *cols*, int *rows*, char **
col_headers)**

Initialize table to starting values.

Parameters:

t,: the table

cols,: the number of columns

rows,: the number of rows
col_headers,: the column titles

Returns:

Returns TRUE on success or FALSE on failure.

Responsible for setting up default values and callbacks withing the table structure

6.70.4.13 Ewl_Widget* ewl_table_new (int cols, int rows, char ** col_headers)

Create a new table.

Parameters:

cols,: the number of columns
rows,: the number of rows
col_headers,: the column titles

Returns:

Returns a a newl table on success, NULL on failure.

6.70.4.14 void ewl_table_reset (Ewl_Table * t, int cols, int rows, char ** col_headers)

Clear the table and set new geometry.

Parameters:

t,: the table
cols,: the new number of columns
rows,: the new number of columns
col_headers,: the new column headers

Returns:

Returns no value

6.70.4.15 void ewl_table_row_h_get (Ewl_Table * table, int row, int * height)

Get the height of a table row.

Parameters:

table,: the table
row,: the row
height,: integer pointer to store the height in

Returns:

Returns no value.

6.70.4.16 void ewl_table_row_h_set (Ewl_Table * *table*, int *row*, int *height*)

Set the height of a table row.

Parameters:

table,: the table
row,: the row
height,: the new height

Returns:

Returns no value.

6.70.4.17 char* ewl_table_selected_get (Ewl_Table * *t*)

Get the text in the current selected box.

Parameters:

t,: the table

Returns:

Returns the text in the currently selected widget in the table

6.70.4.18 unsigned int ewl_table_vhomogeneous_get (Ewl_Table * *table*)

Retrieves the vertical homogeneous flag.

Parameters:

table,: the table to get the vertical layout

Returns:

The vertical homogeneous flag

6.70.4.19 void ewl_table_vhomogeneous_set (Ewl_Table * *table*, unsigned int *h*)

Change the vertical homogeneous layout of the box.

Parameters:

table,: the table to change vertical homogeneous layout
h,: the boolean value to change the vertical layout mode to

Returns:

Returns no value.

Grids use non-homogeneous layout by default, this can be used to change that for vertical orientation, i.e. all rows can have the same height.

6.71 Ewl_Text: A text layout widget

6.71.1 Detailed Description

Defines a class for text layout and formatting.

Remarks:

Inherits from **Ewl_Container** (p. 618).

6.71.2 Tutorial

The **Ewl_Text** (p. 757) widget provides for a multi-line text layout widget. It can be utilized whenever the display of text is required in an application. It works well with the **Ewl_Scrollpane** (p. 732) to provide a scrollable text area.

```
Ewl_Widget *text;
text = ewl_text_new();
ewl_text_text_set(EWL_TEXT(text), "set the text here");
ewl_widget_show(text);
```

Once the text object is created you can change the text, retrieve the current text contents or get the text length. These things can be done with:

```
void ewl_text_text_set(Ewl_Text *t, const char *txt);
void ewl_text_text_prepend(Ewl_Text *t, const char *txt);
void ewl_text_text_append(Ewl_Text *t, const char *txt);
void ewl_text_text_insert(Ewl_Text *t, const char *txt, int len);
char *ewl_text_text_get(Ewl_Text *t);
int ewl_text_length_get(Ewl_Text *t);
void ewl_text_clear(Ewl_Text *t);
void ewl_text_text_delete(Ewl_Text *t, unsigned int len);
```

The **Ewl_Text** (p. 757) widget allows you to perform style changes to the text in the widget. Different portions of the text can be different colours, fonts or styles. You can either set the styling, colours or fonts before the text is set, or you can apply the settings to the text afterwards.

The colour settings of the text can be manipulated with the following:

```
void ewl_text_color_set(Ewl_Text *t, int r, int g, int b, int a);
void ewl_text_color_get(Ewl_Text *t, int *r, int *g, int *b, int *a);
void ewl_text_color_apply(Ewl_Text *t, int r, int g, int b, int a, unsigned int len);
```

There are similar calls to manipulate the font, font size, font colour, background colour, glow colour, outline colour, strikethrough colour, underline colour, double underline colour, alignment, wrap and style information.

Styles have a few extra calls to make them easier to use. These include:

```
void ewl_text_style_add(Ewl_Text *t, Ewl_Text_Style style, unsigned int len);
void ewl_text_style_del(Ewl_Text *t, Ewl_Text_Style style, unsigned int len);
void ewl_text_style_invert(Ewl_Text *t, Ewl_Text_Style style, unsigned int len);
unsigned int ewl_text_style_has(Ewl_Text *t, Ewl_Text_Style style, unsigned int idx);
```

If you want users to be able to select text from the **Ewl_Text** (p. 757) widget you'll use:

```
void ewl_text_selectable_set(Ewl_Text *t, unsigned int selectable);
unsigned int ewl_text_selectable_get(Ewl_Text *t);
```

After the user has made a selection it can be checked and retrieved with:

```
unsigned int ewl_text_has_selection(Ewl_Text *t);
Ewl_Text_Trigger *ewl_text_selection_get(Ewl_Text *t);
char *ewl_text_selection_text_get(Ewl_Text *t);
```

This should hopefully give you some idea of the capabilities of the **Ewl_Text** (p. 757) widget. Take a look at the test code and the header file for more information.

Data Structures

- struct **Ewl_Text**
*Inherits from the **Ewl_Container** (p. 618) class and extends it to provide text layout and formatting.*
- struct **Ewl_Text_Context**
Stores context information for the different formatting nodes.
- struct **Ewl_Text_Trigger**
*Inherits from **Ewl_Widget** (p. 785) and extends to provide a trigger for the text widget.*
- struct **Ewl_Text_Trigger_Area**
*Inherits from **Ewl_Widget** (p. 785) and extends to provide a trigger area.*

Defines

- #define **EWL_TEXT**(x) ((Ewl_Text *)x)
- #define **EWL_TEXT_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TEXT_TYPE))
- #define **EWL_TEXT_SELECTION_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TEXT_SELECTION_TYPE))
- #define **EWL_TEXT_SELECTION_TYPE** "selection"
- #define **EWL_TEXT_TRIGGER**(trigger) ((Ewl_Text_Trigger *) trigger)
- #define **EWL_TEXT_TRIGGER_AREA**(area) ((Ewl_Text_Trigger_Area *) area)
- #define **EWL_TEXT_TRIGGER_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TEXT_TRIGGER_TYPE))
- #define **EWL_TEXT_TRIGGER_TYPE** "trigger"
- #define **EWL_TEXT_TYPE** "text"

Typedefs

- typedef Ewl_Text Ewl_Text
- typedef Ewl_Text_Context Ewl_Text_Context
- typedef Ewl_Text_Trigger Ewl_Text_Trigger
- typedef Ewl_Text_Trigger_Area Ewl_Text_Trigger_Area

Functions

- void **ewl_text_align_apply** (Ewl_Text *t, unsigned int align, unsigned int char_len)
This will set the given alignment from the current cursor position for the given length of text.
- unsigned int **ewl_text_align_get** (Ewl_Text *t, unsigned int char_idx)
Retrieves the alignment value from the given index.
- void **ewl_text_align_set** (Ewl_Text *t, unsigned int align)
Set the current alignment value of the text.
- void **ewl_text_bg_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)
This will set the bg colour of the text from the current cursor position to the given length.
- void **ewl_text_bg_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)
Get the text background colour at the given index.
- void **ewl_text_bg_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)
Set the background colour at the cursor.
- void **ewl_text_cb_child_add** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_text_cb_child_del** (Ewl_Container *c, Ewl_Widget *w, int idx)
- void **ewl_text_cb_configure** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_cb_destroy** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_cb_hide** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_cb_mouse_down** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_cb_mouse_move** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_cb_mouse_up** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_cb_obscure** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_cb_reveal** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_cb_show** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_clear** (Ewl_Text *t)
Clear the text widget.
- void **ewl_text_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)
This will set the given colour from the current cursor position for the specified length.
- void **ewl_text_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)
Retrives the text colour at the given index.
- void **ewl_text_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)
Set the text colour at the cursor.
- void **ewl_text_context_acquire** (Ewl_Text_Context *tx)
- int **ewl_text_context_compare** (Ewl_Text_Context *a, Ewl_Text_Context *b)

- **Ewl_Text_Context * ewl_text_context_default_create (Ewl_Text *t)**
Retrieves the default Ewl_Text_Context (p. 761).
- **Ewl_Text_Context * ewl_text_context_dup (Ewl_Text_Context *old)**
- **int ewl_text_context_init (void)**
- **Ewl_Text_Context * ewl_text_context_new (void)**
- **void ewl_text_context_release (Ewl_Text_Context *tx)**
- **void ewl_text_context_shutdown (void)**
- **unsigned int ewl_text_coord_index_map (Ewl_Text *t, int x, int y)**
Map the given coordinate into an index into the text widget.
- **unsigned int ewl_text_cursor_position_get (Ewl_Text *t)**
Retrieve the cursor position from the text widget.
- **unsigned int ewl_text_cursor_position_line_down_get (Ewl_Text *t)**
Get the index if we were to move the cursor down one line.
- **unsigned int ewl_text_cursor_position_line_up_get (Ewl_Text *t)**
Get the index if we were to move the cursor up one line.
- **void ewl_text_cursor_position_set (Ewl_Text *t, unsigned int char_pos)**
Set the cursor position in the text widget.
- **void ewl_text_double_underline_color_apply (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)**
This will set the double_underline colour of the text from the current cursor position to the given length.
- **void ewl_text_double_underline_color_get (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)**
Retrieve the double underline colour at the given index.
- **void ewl_text_double_underline_color_set (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)**
Set the double underline colour at the cursor.
- **void ewl_text_fmt_dump (Ewl_Text *t)**
Prints out the formatting information for the given text object.
- **void ewl_text_font_apply (Ewl_Text *t, const char *font, unsigned int char_len)**
This will apply the specified font from the current cursor position to the length specified.
- **char * ewl_text_font_get (Ewl_Text *t, unsigned int char_idx)**
This will retrieve the font used at the specified index in the text.
- **void ewl_text_font_set (Ewl_Text *t, const char *font)**
This will set the current font to be used when we insert more text.
- **void ewl_text_font_size_apply (Ewl_Text *t, unsigned int size, unsigned int char_len)**

This will apply the font size to the text from the current cursor position for the given length.

- unsigned int **ewl_text_font_size_get** (Ewl_Text *t, unsigned int char_idx)
Retrieve the font size at the given index.
- void **ewl_text_font_size_set** (Ewl_Text *t, unsigned int size)
Set the font size to use when inserting new text.
- void **ewl_text_font_source_apply** (Ewl_Text *t, const char *source, const char *font, unsigned int char_len)
This will apply the specified font from the current cursor position to the length specified.
- char * **ewl_text_font_source_get** (Ewl_Text *t, unsigned int char_idx)
This will retrieve the font source used at the specified index in the text.
- void **ewl_text_font_source_set** (Ewl_Text *t, const char *source, const char *font)
This will set the current font to be used when we insert more text.
- void **ewl_text_glow_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)
This will set the glow colour of the text from the current cursor position to the given length.
- void **ewl_text_glow_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)
Get the glow colour at the given index.
- void **ewl_text_glow_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)
Set the glow colour at the cursor.
- unsigned int **ewl_text_has_selection** (Ewl_Text *t)
Check if anything is selected in the text widget.
- void **ewl_text_index_geometry_map** (Ewl_Text *t, unsigned int char_idx, int *x, int *y, int *w, int *h)
Map the given character index into a position in the text widget.
- int **ewl_text_init** (Ewl_Text *t)
*Initializes an **Ewl_Text** (p. 757) widget to default values.*
- unsigned int **ewl_text_length_get** (Ewl_Text *t)
Retrieve the character length of the text.
- Ewl_Widget * **ewl_text_new** (void)
*Creates a new **Ewl_Text** (p. 757) widget.*
- void **ewl_text_offsets_get** (Ewl_Text *t, int *x, int *y)
Retrieve the current layout offsets of the text.
- void **ewl_text_offsets_set** (Ewl_Text *t, int x, int y)

Set the current layout offsets of the text.

- void **ewl_text_outline_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)

This will set the outline colour of the text from the current cursor position to the given length.

- void **ewl_text_outline_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)

Get the outline colour at the given index.

- void **ewl_text_outline_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)

Set the outline colour at the cursor.

- unsigned int **ewl_text_selectable_get** (Ewl_Text *t)

Get the selectable state of the text.

- void **ewl_text_selectable_set** (Ewl_Text *t, unsigned int selectable)

Set if the text is selectable.

- Ewl_Text_Trigger * **ewl_text_selection_get** (Ewl_Text *t)

Get the current text selection.

- char * **ewl_text_selection_text_get** (Ewl_Text *t)

Gets the current text of the selection.

- void **ewl_text_shadow_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)

This will set the shadow colour of the text from the current cursor position to the given length.

- void **ewl_text_shadow_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)

Retrieve the shadow colour at the given index.

- void **ewl_text_shadow_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)

Set the shadow colour at the cursor.

- void **ewl_text_strikethrough_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)

This will set the strikethrough colour of the text from the current cursor position to the given length.

- void **ewl_text_strikethrough_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)

Retrieve the strikethrough colour at the given index.

- void **ewl_text_strikethrough_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)

Set the strikethrough colour at the cursor.

- void **ewl_text_style_add** (**Ewl_Text** *t, **Ewl_Text_Style** style, unsigned int char_len)

This will add the given style to the text from the cursor up to length characters.
- void **ewl_text_style_del** (**Ewl_Text** *t, **Ewl_Text_Style** style, unsigned int char_len)

This will delete the given style from the text starting at the cursor up to length characters.
- unsigned int **ewl_text_style_has** (**Ewl_Text** *t, **Ewl_Text_Style** style, unsigned int char_idx)

Check if the given style is set at the given index in the text.
- void **ewl_text_style_invert** (**Ewl_Text** *t, **Ewl_Text_Style** style, unsigned int char_len)

This will invert the given style in the text starting at the cursor up to length characters.
- void **ewl_text_styles_apply** (**Ewl_Text** *t, unsigned int styles, unsigned int char_len)

This will set the given style from the current cursor position for the given length of text.
- unsigned int **ewl_text_styles_get** (**Ewl_Text** *t, unsigned int char_idx)

Retrives the styles in use at the given index.
- void **ewl_text_styles_set** (**Ewl_Text** *t, unsigned int styles)

Sets the given styles into the text at the cursor.
- void **ewl_text_text_append** (**Ewl_Text** *t, const char *text)

Append the text into the text widget.
- void **ewl_text_text_delete** (**Ewl_Text** *t, unsigned int length)

This will delete the specified length of text from the current cursor position.
- char * **ewl_text_text_get** (**Ewl_Text** *t)

Retrieve the text from the text widget.
- void **ewl_text_text_insert** (**Ewl_Text** *t, const char *text, unsigned int char_idx)

Insert the given text into the text widget.
- void **ewl_text_text_prepend** (**Ewl_Text** *t, const char *text)

Prepend the given text into the text widget.
- void **ewl_text_text_set** (**Ewl_Text** *t, const char *text)

Set the text in the text widget.
- int **ewl_text_trigger_area_init** (**Ewl_Text_Trigger_Area** *area, **Ewl_Text_Trigger_Type** type)
- **Ewl_Widget** * **ewl_text_trigger_area_new** (**Ewl_Text_Trigger_Type** type)
- unsigned int **ewl_text_trigger_base_get** (**Ewl_Text_Trigger** *t)

Retrieves the current base position of the cursor.

- void **ewl_text_trigger_base_set** (Ewl_Text_Trigger *t, unsigned int char_pos)
Sets the given position pos as the base for the trigger t.
- void **ewl_text_trigger_cb_destroy** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_trigger_cb_mouse_down** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_trigger_cb_mouse_in** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_trigger_cb_mouse_out** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_trigger_cb_mouse_up** (Ewl_Widget *w, void *ev, void *data)
- int **ewl_text_trigger_init** (Ewl_Text_Trigger *trigger, Ewl_Text_Trigger_Type type)
Initializes a trigger to default values.
- unsigned int **ewl_text_trigger_length_get** (Ewl_Text_Trigger *t)
Retrieves the length from the cursor t.
- void **ewl_text_trigger_length_set** (Ewl_Text_Trigger *t, unsigned int char_len)
Sets the length len on the trigger t.
- Ewl_Text_Trigger * **ewl_text_trigger_new** (Ewl_Text_Trigger_Type type)
Creates a new trigger for the text object.
- unsigned int **ewl_text_trigger_start_pos_get** (Ewl_Text_Trigger *t)
Retrieves the start position of the trigger.
- void **ewl_text_trigger_start_pos_set** (Ewl_Text_Trigger *t, unsigned int char_pos)
Sets the start position of the trigger t to position pos.
- Ewl_Text_Trigger_Type **ewl_text_trigger_type_get** (Ewl_Text_Trigger *t)
Retrieves the type of the trigger.
- void **ewl_text_triggers_configure** (Ewl_Text *t)
- void **ewl_text_triggers_hide** (Ewl_Text *t)
- void **ewl_text_triggers_realize** (Ewl_Text *t)
- void **ewl_text_triggers_show** (Ewl_Text *t)
- void **ewl_text_triggers_unrealize** (Ewl_Text *t)
- void **ewl_text_underline_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)
This will set the underline colour of the text from the current cursor position to the given length.
- void **ewl_text_underline_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)
Retrieve the underline colour at the given index.
- void **ewl_text_underline_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)
Set the underline colour at the cursor.
- void **ewl_text_wrap_apply** (Ewl_Text *t, Ewl_Text_Wrap wrap, unsigned int char_len)

This will apply the given wrap value from the current cursor position for the given length of text.

- **Ewl_Text_Wrap** **ewl_text_wrap_get** (**Ewl_Text** *t, unsigned int char_idx)
Retrives the text wrap value at the given index.
- void **ewl_text_wrap_set** (**Ewl_Text** *t, **Ewl_Text_Wrap** wrap)
Sets the wrap value of the text at the given index.

6.71.3 Define Documentation

6.71.3.1 **#define EWL_TEXT(x) ((Ewl_Text *)x)**

Typecasts a pointer to an **Ewl_Text** (p. 757) pointer.

6.71.3.2 **#define EWL_TEXT_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TEXT_TYPE))**

Returns TRUE if the widget is an **Ewl_Text** (p. 757), FALSE otherwise

6.71.3.3 **#define EWL_TEXT_SELECTION_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TEXT_SELECTION_TYPE))**

Returns TRUE if the widget is an **Ewl_Text_Selection**, FALSE otherwise

6.71.3.4 **#define EWL_TEXT_SELECTION_TYPE "selection"**

The type name for the **Ewl_Text_Selection** widget

6.71.3.5 **#define EWL_TEXT_TRIGGER(trigger) ((Ewl_Text_Trigger *) trigger)**

Typecasts a pointer to an **Ewl_Text_Trigger** (p. 765) pointer

6.71.3.6 **#define EWL_TEXT_TRIGGER_AREA(area) ((Ewl_Text_Trigger_Area *) area)**

Typecasts a pointer to an **Ewl_Text_Trigger_Area** (p. 767) pointer

6.71.3.7 **#define EWL_TEXT_TRIGGER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TEXT_TRIGGER_TYPE))**

Returns TRUE if the widget is an **Ewl_Text_Trigger** (p. 765), FALSE otherwise

6.71.3.8 **#define EWL_TEXT_TRIGGER_TYPE "trigger"**

The type name for the **Ewl_Text_Trigger** (p. 765) widget

6.71.3.9 `#define EWL_TEXT_TYPE "text"`

The type name for the `Ewl_Text` (p. 757) widget

6.71.4 Typedef Documentation

6.71.4.1 `typedef struct Ewl_Text Ewl_Text`

Provides for layout of text as well as formatting portions of the text in different ways.

6.71.4.2 `typedef struct Ewl_Text_Context Ewl_Text_Context`

A context node for formatting information

6.71.4.3 `typedef struct Ewl_Text_Trigger Ewl_Text_Trigger`

A trigger in the text, used for selections and clickable text

6.71.4.4 `typedef struct Ewl_Text_Trigger_Area Ewl_Text_Trigger_Area`

Typdef for the `Ewl_Text_Trigger_Area` (p. 767) struct

6.71.5 Function Documentation

6.71.5.1 `void ewl_text_align_apply (Ewl_Text * t, unsigned int align, unsigned int char_len)`

This will set the given alignment from the current cursor position for the given length of text.

Parameters:

- t*,: The `Ewl_Text` (p. 757) to apply the alignment too
- align*,: The alignment to apply
- char_len*,: The length to apply the alignment for

Returns:

Returns no value

6.71.5.2 `unsigned int ewl_text_align_get (Ewl_Text * t, unsigned int char_idx)`

Retrieves the alignment value from the given index.

Parameters:

- t*,: The `Ewl_Text` (p. 757) to get the alignment from
- char_idx*,: The index to get the alignment from

Returns:

Returns the current text alignment value

6.71.5.3 void ewl_text_align_set (Ewl_Text * *t*, unsigned int *align*)

Set the current alignment value of the text.

Parameters:

t,: The **Ewl_Text** (p. 757) ot set the alignment into
align,: The alignment to set

Returns:

Returns no value

6.71.5.4 void ewl_text_bg_color_apply (Ewl_Text * *t*, unsigned int *r*, unsigned int *g*, unsigned int *b*, unsigned int *a*, unsigned int *char_len*)

This will set the bg colour of the text from the current cursor position to the given length.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text background colour of
r,: The red value
g,: The green value
b,: The blue value
a,: The alpha value
char_len,: The length of text to apply the bg colour over

Returns:

Returns no value

6.71.5.5 void ewl_text_bg_color_get (Ewl_Text * *t*, unsigned int * *r*, unsigned int * *g*, unsigned int * *b*, unsigned int * *a*, unsigned int *char_idx*)

Get the text background colour at the given index.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the text background colour from
r,: Where to put the red value
g,: Where to put the green value
b,: Where to put the blue value
a,: Where to put the alpha value
char_idx,: The index to get the colour from

Returns:

Returns no value

6.71.5.6 void ewl_text_bg_color_set (Ewl_Text * *t*, unsigned int *r*, unsigned int *g*, unsigned int *b*, unsigned int *a*)

Set the background colour at the cursor.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text background colour of
r,: The red value
g,: The green value
b,: The blue value
a,: The alpha value

Returns:

Returns no value

6.71.5.7 void ewl_text_cb_child_add (Ewl_Container * *c*, Ewl_Widget * *w*)

6.71.5.8 void ewl_text_cb_child_del (Ewl_Container * *c*, Ewl_Widget * *w*, int *idx*)

6.71.5.9 void ewl_text_cb_configure (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.71.5.10 void ewl_text_cb_destroy (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.71.5.11 void ewl_text_cb_hide (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.71.5.12 void ewl_text_cb_mouse_down (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.71.5.13 void ewl_text_cb_mouse_move (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.71.5.14 void ewl_text_cb_mouse_up (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.71.5.15 void ewl_text_cb_obscure (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.71.5.16 void ewl_text_cb_reveal (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.71.5.17 void ewl_text_cb_show (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.71.5.18 void ewl_text_clear (Ewl_Text * *t*)

Clear the text widget.

Parameters:

t,: The **Ewl_Text** (p. 757) to clear

Returns:

Returns no value

6.71.5.19 `void ewl_text_color_apply (Ewl_Text * t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)`

This will set the given colour from the current cursor position for the specified length.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the colour into
r,: The red value to set
g,: The green value to set
b,: The blue value to set
a,: The alpha value to set
char_len,: The length of text to apply the colour over

Returns:

Returns no value

6.71.5.20 `void ewl_text_color_get (Ewl_Text * t, unsigned int * r, unsigned int * g, unsigned int * b, unsigned int * a, unsigned int char_idx)`

Retrives the text colour at the given index.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the colour from
r,: Where to put the red value
g,: Where to put the green value
b,: Where to put the blue value
a,: Where to put the alpha value
char_idx,: The index to get the colour from

Returns:

Returns no value

6.71.5.21 `void ewl_text_color_set (Ewl_Text * t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)`

Set the text colour at the cursor.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the colour on
r,: The red value
g,: The green value
b,: The blue value
a,: The alpha value

Returns:

Returns no value

6.71.5.22 void ewl_text_context_acquire (Ewl_Text_Context * *tx*)

6.71.5.23 int ewl_text_context_compare (Ewl_Text_Context * *a*,
Ewl_Text_Context * *b*)

6.71.5.24 Ewl_Text_Context* ewl_text_context_default_create (Ewl_Text * *t*)

Retrieves the default **Ewl_Text_Context** (p. 761).

Parameters:

t,: The **Ewl_Text** (p. 757) to work with

Returns:

Returns the default **Ewl_Text_Context** (p. 761)

6.71.5.25 Ewl_Text_Context* ewl_text_context_dup (Ewl_Text_Context * *old*)

6.71.5.26 int ewl_text_context_init (void)

6.71.5.27 Ewl_Text_Context* ewl_text_context_new (void)

6.71.5.28 void ewl_text_context_release (Ewl_Text_Context * *tx*)

6.71.5.29 void ewl_text_context_shutdown (void)

6.71.5.30 unsigned int ewl_text_coord_index_map (Ewl_Text * *t*, int *x*, int *y*)

Map the given coordinate into an index into the text widget.

Parameters:

t,: The **Ewl_Text** (p. 757) to mapp the coords into

x,: The x coord to map

y,: The y coord to map

Returns:

Returns the character index of the given coordinates

6.71.5.31 unsigned int ewl_text_cursor_position_get (Ewl_Text * *t*)

Retrieve the cursor position from the text widget.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the cursor position from

Returns:

Returns the current cursor position in the widget

6.71.5.32 unsigned int ewl_text_cursor_position_line_down_get (Ewl_Text * *t*)

Get the index if we were to move the cursor down one line.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the cursor position one line down from

Returns:

Returns the cursor position if we moved down one line

6.71.5.33 unsigned int ewl_text_cursor_position_line_up_get (Ewl_Text * *t*)

Get the index if we were to move the cursor up one line.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the cursor position one line up from

Returns:

Returns the cursor position if we moved up one line

6.71.5.34 void ewl_text_cursor_position_set (Ewl_Text * *t*, unsigned int *char_pos*)

Set the cursor position in the text widget.

Parameters:

t,: The **Ewl_Text** (p. 757) widget to set the position into

char_pos,: The position to set

Returns:

Returns no value.

6.71.5.35 void ewl_text_double_underline_color_apply (Ewl_Text * *t*, unsigned int *r*, unsigned int *g*, unsigned int *b*, unsigned int *a*, unsigned int *char_len*)

This will set the double_underline colour of the text from the current cursor position to the given length.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text double underline colour of

r,: The red value

g,: The green value

b,: The blue value

a,: The alpha value

char_len,: The length of text to apply the double underline colour over

Returns:

Returns no value

6.71.5.36 `void ewl_text_double_underline_color_get (Ewl_Text * t, unsigned int * r, unsigned int * g, unsigned int * b, unsigned int * a, unsigned int char_idx)`

Retrieve the double underline colour at the given index.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the text double underline colour from

r,: Where to put the red value

g,: Where to put the green value

b,: Where to put the blue value

a,: Where to put the alpha value

char_idx,: The index to get the colour from

Returns:

Returns no value

6.71.5.37 `void ewl_text_double_underline_color_set (Ewl_Text * t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)`

Set the double underline colour at the cursor.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text double underline colour of

r,: The red value

g,: The green value

b,: The blue value

a,: The alpha value

Returns:

Returns no value

6.71.5.38 `void ewl_text_fmt_dump (Ewl_Text * t)`

Prints out the formatting information for the given text object.

Parameters:

t,: The text to dump the formatting from

Returns:

Returns no value

6.71.5.39 `void ewl_text_font_apply (Ewl_Text * t, const char * font, unsigned int char_len)`

This will apply the specified *font* from the current cursor position to the length specified.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the font too
font,: The font to set
char_len,: The distance to set the font over

Returns:

Returns no value

6.71.5.40 `char* ewl_text_font_get (Ewl_Text * t, unsigned int char_idx)`

This will retrieve the font used at the specified index in the text.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the font from
char_idx,: The index to get the font at

Returns:

Returns no value

6.71.5.41 `void ewl_text_font_set (Ewl_Text * t, const char * font)`

This will set the current font to be used when we insert more text.

Parameters:

t,: The **Ewl_Widget** (p. 785) to set the font into
font,: The font to set

Returns:

Returns no value

6.71.5.42 `void ewl_text_font_size_apply (Ewl_Text * t, unsigned int size, unsigned int char_len)`

This will apply the font size to the text from the current cursor position for the given length.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the size into
size,: The size to set
char_len,: Length of block to get the new size

Returns:

Returns no value

6.71.5.43 unsigned int ewl_text_font_size_get (Ewl_Text * *t*, unsigned int *char_idx*)

Retrieve the font size at the given index.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the size from
char_idx,: The index you want to get the size for

Returns:

Returns no value

6.71.5.44 void ewl_text_font_size_set (Ewl_Text * *t*, unsigned int *size*)

Set the font size to use when inserting new text.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the size too
size,: The size to set the font too

Returns:

Returns no value

6.71.5.45 void ewl_text_font_source_apply (Ewl_Text * *t*, const char * *source*, const char * *font*, unsigned int *char_len*)

This will apply the specfied *font* from the current cursor position to the length specified.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the font too
source,: The font souce
font,: The font to set
char_len,: The distance to set the font over

Returns:

Returns no value

6.71.5.46 char* ewl_text_font_source_get (Ewl_Text * *t*, unsigned int *char_idx*)

This will retrive the font source used at the specified index in the text.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the font from
char_idx,: The index to get the font at

Returns:

Returns no value

6.71.5.47 `void ewl_text_font_source_set (Ewl_Text * t, const char * source,
const char * font)`

This will set the current font to be used when we insert more text.

Parameters:

t,: The **Ewl_Widget** (p. 785) to set the font into

source,: The font source to set

font,: The font to set

Returns:

Returns no value

6.71.5.48 `void ewl_text_glow_color_apply (Ewl_Text * t, unsigned int r,
unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)`

This will set the glow colour of the text from the current cursor position to the given length.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text glow colour of

r,: The red value

g,: The green value

b,: The blue value

a,: The alpha value

char_len,: The length of text to apply the glow colour over

Returns:

Returns no value

6.71.5.49 `void ewl_text_glow_color_get (Ewl_Text * t, unsigned int * r,
unsigned int * g, unsigned int * b, unsigned int * a, unsigned int
char_idx)`

Get the glow colour at the given index.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the text glow colour from

r,: Where to put the red value

g,: Where to put the green value

b,: Where to put the blue value

a,: Where to put the alpha value

char_idx,: The index to get the colour from

Returns:

Returns no value

6.71.5.50 `void ewl_text_glow_color_set (Ewl_Text * t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)`

Set the glow colour at the cursor.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text glow colour of
r,: The red value
g,: The green value
b,: The blue value
a,: The alpha value

Returns:

Returns no value

6.71.5.51 `unsigned int ewl_text_has_selection (Ewl_Text * t)`

Check if anything is selected in the text widget.

Parameters:

t,: The text to check if there is a selection

Returns:

Returns TRUE if there is selected text, FALSE otherwise

6.71.5.52 `void ewl_text_index_geometry_map (Ewl_Text * t, unsigned int char_idx, int * x, int * y, int * w, int * h)`

Map the given character index into a position in the text widget.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the geometry from
char_idx,: The character index to get the geometry for
x,: Where to put the x value
y,: Where to put the y value
w,: Where to put the w value
h,: Where to put the h value

6.71.5.53 `int ewl_text_init (Ewl_Text * t)`

Initializes an **Ewl_Text** (p. 757) widget to default values.

Parameters:

t,: The **Ewl_Text** (p. 757) widget

Returns:

Returns TRUE on successfully init or FALSE on failure

6.71.5.54 unsigned int ewl_text_length_get (Ewl_Text * *t*)

Retrieve the character length of the text.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the length from

Returns:

Returns the character length of the text in the widget *t*

6.71.5.55 Ewl_Widget* ewl_text_new (void)

Creates a new **Ewl_Text** (p. 757) widget.

Returns:

Returns a new **Ewl_Text** (p. 757) widget on success, NULL on failure.

6.71.5.56 void ewl_text_offsets_get (Ewl_Text * *t*, int * *x*, int * *y*)

Retrieve the current layout offsets of the text.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the layout offsets from

x,: A pointer to an integer to store the x offset of the text display

y,: A pointer to an integer to store the y offset of the text display

Returns:

Returns no value.

6.71.5.57 void ewl_text_offsets_set (Ewl_Text * *t*, int *x*, int *y*)

Set the current layout offsets of the text.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the layout offsets

x,: The x amount to offset of the text display

y,: The y amount to offset of the text display

Returns:

Returns no value.

6.71.5.58 void ewl_text_outline_color_apply (Ewl_Text * *t*, unsigned int *r*, unsigned int *g*, unsigned int *b*, unsigned int *a*, unsigned int *char_len*)

This will set the outline colour of the text from the current cursor position to the given length.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text outline colour of
r,: The red value
g,: The green value
b,: The blue value
a,: The alpha value
char_len,: The length of text to apply the outline colour over

Returns:

Returns no value

6.71.5.59 void ewl_text_outline_color_get (Ewl_Text * *t*, unsigned int * *r*, unsigned int * *g*, unsigned int * *b*, unsigned int * *a*, unsigned int *char_idx*)

Get the outline colour at the given index.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the text outline colour from
r,: Where to put the red value
g,: Where to put the green value
b,: Where to put the blue value
a,: Where to put the alpha value
char_idx,: The index to get the colour from

Returns:

Returns no value

6.71.5.60 void ewl_text_outline_color_set (Ewl_Text * *t*, unsigned int *r*, unsigned int *g*, unsigned int *b*, unsigned int *a*)

Set the outline colour at the cursor.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text outline colour of
r,: The red value
g,: The green value
b,: The blue value
a,: The alpha value

Returns:

Returns no value

6.71.5.61 `unsigned int ewl_text_selectable_get (Ewl_Text * t)`

Get the selectable state of the text.

Parameters:

t,: The text to get the selectable value from

Returns:

Returns the selectable value of the widget

6.71.5.62 `void ewl_text_selectable_set (Ewl_Text * t, unsigned int selectable)`

Set if the text is selectable.

Parameters:

t,: The text to set the selectable value of

selectable,: The selectable value to set

Returns:

Returns no value

6.71.5.63 `Ewl_Text_Trigger* ewl_text_selection_get (Ewl_Text * t)`

Get the current text selection.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the selection from

Returns:

Returns the selection object of this text or NULL if no current selection

6.71.5.64 `char* ewl_text_selection_text_get (Ewl_Text * t)`

Gets the current text of the selection.

Parameters:

t,: The **Ewl_Text** (p. 757) widget to get the selection text from

Returns:

Returns the selection text or NULL if none set

6.71.5.65 void ewl_text_shadow_color_apply (Ewl_Text * *t*, unsigned int *r*, unsigned int *g*, unsigned int *b*, unsigned int *a*, unsigned int *char_len*)

This will set the shadow colour of the text from the current cursor position to the given length.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text shadow colour of
r,: The red value
g,: The green value
b,: The blue value
a,: The alpha value
char_len,: The length of text to apply the shadow colour over

Returns:

Returns no value

6.71.5.66 void ewl_text_shadow_color_get (Ewl_Text * *t*, unsigned int * *r*, unsigned int * *g*, unsigned int * *b*, unsigned int * *a*, unsigned int *char_idx*)

Retrieve the shadow colour at the given index.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the text shadow colour from
r,: Where to put the red value
g,: Where to put the green value
b,: Where to put the blue value
a,: Where to put the alpha value
char_idx,: The index to get the colour from

Returns:

Returns no value

6.71.5.67 void ewl_text_shadow_color_set (Ewl_Text * *t*, unsigned int *r*, unsigned int *g*, unsigned int *b*, unsigned int *a*)

Set the shadow colour at the cursor.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text shadow colour of
r,: The red value
g,: The green value
b,: The blue value
a,: The alpha value

Returns:

Returns no value

6.71.5.68 `void ewl_text_strikethrough_color_apply (Ewl_Text * t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)`

This will set the strikethrough colour of the text from the current cursor position to the given length.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text strikethrough colour of
r,: The red value
g,: The green value
b,: The blue value
a,: The alpha value
char_len,: The length of text to apply the strikethrough colour over

Returns:

Returns no value

6.71.5.69 `void ewl_text_strikethrough_color_get (Ewl_Text * t, unsigned int * r, unsigned int * g, unsigned int * b, unsigned int * a, unsigned int char_idx)`

Retrieve the strikethrough colour at the given index.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the text strikethrough colour from
r,: Where to put the red value
g,: Where to put the green value
b,: Where to put the blue value
a,: Where to put the alpha value
char_idx,: The index to get the colour from

Returns:

Returns no value

6.71.5.70 `void ewl_text_strikethrough_color_set (Ewl_Text * t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)`

Set the strikethrough colour at the cursor.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text strikethrough colour of
r,: The red value
g,: The green value
b,: The blue value

a,: The alpha value

Returns:

Returns no value

6.71.5.71 void ewl_text_style_add (Ewl_Text * *t*, Ewl_Text_Style *style*, unsigned int *char_len*)

This will add the given style to the text from the cursor up to length characters.

Parameters:

t,: The text to add the style too

style,: The style to add to the text

char_len,: The length of text to add the style too

Returns:

Returns no value

6.71.5.72 void ewl_text_style_del (Ewl_Text * *t*, Ewl_Text_Style *style*, unsigned int *char_len*)

This will delete the given style from the text starting at the cursor up to length characters.

Parameters:

t,: The text to delete the style from

style,: The style to delete from the text

char_len,: The length of text to delete the style from

Returns:

Returns no value

6.71.5.73 unsigned int ewl_text_style_has (Ewl_Text * *t*, Ewl_Text_Style *style*, unsigned int *char_idx*)

Check if the given style is set at the given index in the text.

Parameters:

t,: The text to check for the style

style,: The style to check for

char_idx,: The index to check for the style

Returns:

Returns no value

6.71.5.74 `void ewl_text_style_invert (Ewl_Text * t, Ewl_Text_Style style, unsigned int char_len)`

This will invert the given style in the text starting at the cursor up to length characters.

Parameters:

t,: The text to invert the style on
style,: The style to invert in the text
char_len,: The length of text to invert the style on

Returns:

Returns no value

6.71.5.75 `void ewl_text_styles_apply (Ewl_Text * t, unsigned int styles, unsigned int char_len)`

This will set the given style from the current cursor position for the given length of text.

Parameters:

t,: The **Ewl_Text** (p. 757) to apply the style too
styles,: The styles to set into the text
char_len,: The length of text to apply the style too

Returns:

Returns no value

6.71.5.76 `unsigned int ewl_text_styles_get (Ewl_Text * t, unsigned int char_idx)`

Retrieves the styles in use at the given index.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the style from
char_idx,: The index to get the style from

Returns:

Get the styles set at the given index in the text

6.71.5.77 `void ewl_text_styles_set (Ewl_Text * t, unsigned int styles)`

Sets the given styles into the text at the cursor.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the style into
styles,: The styles to set into the text

Returns:

Returns no value

6.71.5.78 void ewl_text_text_append (Ewl_Text * *t*, const char * *text*)

Append the text into the text widget.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text into
text,: The text to set into the widget

Returns:

Returns no value

6.71.5.79 void ewl_text_text_delete (Ewl_Text * *t*, unsigned int *char_len*)

This will delete the specified length of text from the current cursor position.

Parameters:

t,: The **Ewl_Text** (p. 757) to delete the text from
char_len,: The length of text to delete

Returns:

Returns no value

6.71.5.80 char* ewl_text_text_get (Ewl_Text * *t*)

Retrieve the text from the text widget.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the text from

Returns:

Returns the text in the widget *t* or NULL if no text is set

6.71.5.81 void ewl_text_text_insert (Ewl_Text * *t*, const char * *text*, unsigned int *char_idx*)

Insert the given text into the text widget.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text into
text,: The text to set into the widget
char_idx,: The index to insert the text at

Returns:

Returns no value

6.71.5.82 `void ewl_text_text_prepend (Ewl_Text * t, const char * text)`

Prepend the given text into the text widget.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text into

text,: The text to set into the widget

Returns:

Returns no value

6.71.5.83 `void ewl_text_text_set (Ewl_Text * t, const char * text)`

Set the text in the text widget.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text into

text,: The text to set into the widget

Returns:

Returns no value

6.71.5.84 `int ewl_text_trigger_area_init (Ewl_Text_Trigger_Area * area,
Ewl_Text_Trigger_Type type)`**6.71.5.85** `Ewl_Widget* ewl_text_trigger_area_new (Ewl_Text_Trigger_Type
type)`**6.71.5.86** `unsigned int ewl_text_trigger_base_get (Ewl_Text_Trigger * t)`

Retrieves the current base position of the cursor.

Parameters:

t,: The trigger to work with

Returns:

Returns the current base position of the cursor

6.71.5.87 `void ewl_text_trigger_base_set (Ewl_Text_Trigger * t, unsigned int
char_pos)`

Sets the given position *pos* as the base for the trigger *t*.

Parameters:

t,: The trigger to work with

char_pos,: The position to set as the base for the cursor

Returns:

Returns no value

6.71.5.88 void ewl_text_trigger_cb_destroy (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.71.5.89 void ewl_text_trigger_cb_mouse_down (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.71.5.90 void ewl_text_trigger_cb_mouse_in (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.71.5.91 void ewl_text_trigger_cb_mouse_out (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.71.5.92 void ewl_text_trigger_cb_mouse_up (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.71.5.93 int ewl_text_trigger_init (Ewl_Text_Trigger * *trigger*, Ewl_Text_Trigger_Type *type*)

Initializes a trigger to default values.

Parameters:

trigger,: The trigger to initialize

type,: The type of the trigger

Returns:

Returns TRUE if successful of FALSE otherwise

6.71.5.94 unsigned int ewl_text_trigger_length_get (Ewl_Text_Trigger * *t*)

Retrieves the length from the cursor *t*.

Parameters:

t,: The trigger to work with

Returns:

Returns the length of the trigger

6.71.5.95 void ewl_text_trigger_length_set (Ewl_Text_Trigger * *t*, unsigned int *char_len*)

Sets the length *len* on the trigger *t*.

Parameters:

t,: The trigger to work with
char_len,: The length to set on the cursor

Returns:

Returns no value

6.71.5.96 Ewl_Text_Trigger* ewl_text_trigger_new (Ewl_Text_Trigger_Type *type*)

Creates a new trigger for the text object.

Parameters:

type,: The type of trigger to create

Returns:

Returns a new ewl_text_trigger widget

6.71.5.97 unsigned int ewl_text_trigger_start_pos_get (Ewl_Text_Trigger * *t*)

Retrieves the start position of the trigger.

Parameters:

t,: The trigger to work with

Returns:

Returns the current start position of the trigger

6.71.5.98 void ewl_text_trigger_start_pos_set (Ewl_Text_Trigger * *t*, unsigned int *char_pos*)

Sets the start position of the trigger *t* to position *pos*.

Parameters:

t,: The trigger to work with
char_pos,: The position to set

Returns:

Returns no value

6.71.5.99 Ewl_Text_Trigger_Type ewl_text_trigger_type_get (Ewl_Text_Trigger * *t*)

Retrieves the type of the trigger.

Parameters:

t,: The trigger to work with

Returns:

Returns the type of the trigger

6.71.5.100 void ewl_text_triggers_configure (Ewl_Text * *t*)

6.71.5.101 void ewl_text_triggers_hide (Ewl_Text * *t*)

6.71.5.102 void ewl_text_triggers_realize (Ewl_Text * *t*)

6.71.5.103 void ewl_text_triggers_show (Ewl_Text * *t*)

6.71.5.104 void ewl_text_triggers_unrealize (Ewl_Text * *t*)

6.71.5.105 void ewl_text_underline_color_apply (Ewl_Text * *t*, unsigned int *r*, unsigned int *g*, unsigned int *b*, unsigned int *a*, unsigned int *char_len*)

This will set the underline colour of the text from the current cursor position to the given length.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text underline colour of

r,: The red value

g,: The green value

b,: The blue value

a,: The alpha value

char_len,: The length of text to apply the underline colour over

Returns:

Returns no value

6.71.5.106 void ewl_text_underline_color_get (Ewl_Text * *t*, unsigned int * *r*, unsigned int * *g*, unsigned int * *b*, unsigned int * *a*, unsigned int *char_idx*)

Retrieve the underline colour at the given index.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the text underline colour from

r,: Where to put the red value

g,: Where to put the green value
b,: Where to put the blue value
a,: Where to put the alpha value
char_idx,: The index to get the colour from

Returns:

Returns no value

6.71.5.107 `void ewl_text_underline_color_set (Ewl_Text * t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)`

Set the underline colour at the cursor.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the text underline colour of
r,: The red value
g,: The green value
b,: The blue value
a,: The alpha value

Returns:

Returns no value

6.71.5.108 `void ewl_text_wrap_apply (Ewl_Text * t, Ewl_Text_Wrap wrap, unsigned int char_len)`

This will apply the given wrap value from the current cursor position for the given length of text.

Parameters:

t,: The **Ewl_Text** (p. 757) to apply the wrap value too
wrap,: The wrap value to apply
char_len,: The length of text to apply the wrap value over

Returns:

Returns no value

6.71.5.109 `Ewl_Text_Wrap ewl_text_wrap_get (Ewl_Text * t, unsigned int char_idx)`

Retrives the text wrap value at the given index.

Parameters:

t,: The **Ewl_Text** (p. 757) to get the wrap value for
char_idx,: The index to get the wrap value from

Returns:

Returns the wrap value of the text at the given index

6.71.5.110 void ewl_text_wrap_set (Ewl_Text * *t*, Ewl_Text_Wrap *wrap*)

Sets the wrap value of the text at the given index.

Parameters:

t,: The **Ewl_Text** (p. 757) to set the wrap into
wrap,: The wrap value to set

Returns:

Returns no value

6.72 Ewl_Theme: Methods for Accessing and Modifying Theme Data

6.72.1 Detailed Description

Provides methods for accessing theme data, global theme data or per-widget theme data.

Defines

- `#define EWL_THEME_KEY_NOMATCH ((char *)0xdeadbeef)`

Functions

- `int ewl_theme_data_int_get (Ewl_Widget *w, char *k)`
Retrieve an integer value from a widgets theme.
- `void ewl_theme_data_int_set (Ewl_Widget *w, char *k, int v)`
Store data into a widgets theme.
- `void ewl_theme_data_reset (Ewl_Widget *w)`
Reset the theme settings for the widget w.
- `const char * ewl_theme_data_str_get (Ewl_Widget *w, char *k)`
Retrieve an string value from a widgets theme.
- `void ewl_theme_data_str_set (Ewl_Widget *w, char *k, char *v)`
Store data into a widgets theme.
- `void ewl_theme_font_path_add (char *path)`
Add a specified path to the font search path.
- `Ecore_List * ewl_theme_font_path_get (void)`
retrieve the path of a widgets theme's fonts
- `char * ewl_theme_image_get (Ewl_Widget *w, char *k)`
retrieve the path to an image from a widgets theme
- `int ewl_theme_init (void)`
- `const char * ewl_theme_path_get (void)`
Return the path of the current theme.
- `void ewl_theme_shutdown (void)`
- `int ewl_theme_theme_set (const char *theme)`
- `int ewl_theme_widget_init (Ewl_Widget *w)`
- `void ewl_theme_widget_shutdown (Ewl_Widget *w)`

6.72.2 Define Documentation

6.72.2.1 `#define EWL_THEME_KEY_NOMATCH ((char *)0xdeadbeef)`

6.72.3 Function Documentation

6.72.3.1 `int ewl_theme_data_int_get (Ewl_Widget * w, char * k)`

Retrieve an integer value from a widgets theme.

Parameters:

w,: the widget to search

k,: the key to search for

Returns:

Returns the integer associated with key *k* on success, 0 on failure.

6.72.3.2 `void ewl_theme_data_int_set (Ewl_Widget * w, char * k, int v)`

Store data into a widgets theme.

Parameters:

w,: the widget to change theme data

k,: the key to change

v,: the data to assign to the key

Returns:

Returns no value.

Changes the theme data in widget *w* so that key *k* now is associated with value *v*.

6.72.3.3 `void ewl_theme_data_reset (Ewl_Widget * w)`

Reset the theme settings for the widget *w*.

Parameters:

w,: the widget to reset theme settings

Returns:

Returns no value.

6.72.3.4 `const char* ewl_theme_data_str_get (Ewl_Widget * w, char * k)`

Retrieve an string value from a widgets theme.

Parameters:

w,: the widget to search

k,: the key to search for

Returns:

Returns the string associated with *k* on success, NULL on failure.

6.72.3.5 void ewl_theme_data_str_set (Ewl_Widget * *w*, char * *k*, char * *v*)

Store data into a widgets theme.

Parameters:

w,: the widget to change theme data

k,: the key to change

v,: the data to assign to the key

Returns:

Returns no value.

Changes the theme data in widget *w* so that key *k* now is associated with value *v*.

6.72.3.6 void ewl_theme_font_path_add (char * *path*)

Add a specified path to the font search path.

Parameters:

path,: the font to add to the search path

Returns:

Returns no value.

Duplicates the string pointed to by *path* and adds it to the list of paths that are searched for fonts.

6.72.3.7 Ecore_List* ewl_theme_font_path_get (void)

retrieve the path of a widgets theme's fonts

Returns:

Returns the font path of widget *w* on success, NULL on failure.

6.72.3.8 char* ewl_theme_image_get (Ewl_Widget * *w*, char * *k*)

retrieve the path to an image from a widgets theme

Parameters:

w,: the widget to search

k,: the image to search for

Returns:

Returns the path of image key *k* on success, NULL on failure.

6.72.3.9 int ewl_theme_init (void)**6.72.3.10 const char* ewl_theme_path_get (void)**

Return the path of the current theme.

Returns:

Returns the current theme path on success, NULL on failure

6.72.3.11 void ewl_theme_shutdown (void)**6.72.3.12 int ewl_theme_theme_set (const char * *theme*)****6.72.3.13 int ewl_theme_widget_init (Ewl_Widget * *w*)****6.72.3.14 void ewl_theme_widget_shutdown (Ewl_Widget * *w*)**

6.73 Ewl_Toolbar: A toolbar widget

6.73.1 Detailed Description

Defines a widget to contain a toolbar.

Remarks:

Inherits from **Ewl_Menubar** (p. 701).

Data Structures

- struct **Ewl_Toolbar**

*Inherits from **Ewl_Menubar** (p. 701) and extends to provide a toolbar layout.*

Defines

- #define **EWL_TOOLBAR**(toolbar) ((**Ewl_Toolbar** *) toolbar)
- #define **EWL_TOOLBAR_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TOOLBAR_TYPE))
- #define **EWL_TOOLBAR_TYPE** "toolbar"

Typedefs

- typedef **Ewl_Toolbar** **Ewl_Toolbar**

Functions

- **Ewl_Widget** * **ewl_htoolbar_new** (void)
Allocate a new toolbar widget with horizontal orientation.
- void **ewl_toolbar_cb_child_add** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_toolbar_icon_part_hide** (**Ewl_Toolbar** *t, **Ewl_Icon_Part** part)
Set the orientation of the toolbar.
- int **ewl_toolbar_init** (**Ewl_Toolbar** *t)
Initialize the toolbar and inherited fields.
- **Ewl_Widget** * **ewl_toolbar_new** (void)
Allocate a new toolbar widget with default (horizontal) orientation.
- **Ewl_Orientation** **ewl_toolbar_orientation_get** (**Ewl_Toolbar** *t)
This retrieves the current orientation of the toolbar.
- void **ewl_toolbar_orientation_set** (**Ewl_Toolbar** *t, **Ewl_Orientation** o)
Set the orientation of the toolbar.
- **Ewl_Widget** * **ewl_vtoolbar_new** (void)
Allocate a new toolbar widget with vertical orientation.

6.73.2 Define Documentation

6.73.2.1 `#define EWL_TOOLBAR(toolbar) ((Ewl_Toolbar *) toolbar)`

Typecast a pointer to an **Ewl_Toolbar** (p. 768) pointer.

6.73.2.2 `#define EWL_TOOLBAR_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TOOLBAR_TYPE))`

Returns TRUE if the widget is an **Ewl_Toolbar** (p. 768), FALSE otherwise

6.73.2.3 `#define EWL_TOOLBAR_TYPE "toolbar"`

The type name for the **Ewl_Toolbar** (p. 768) widget

6.73.3 Typedef Documentation

6.73.3.1 `typedef struct Ewl_Toolbar Ewl_Toolbar`

Ewl_Toolbar (p. 768)

6.73.4 Function Documentation

6.73.4.1 `Ewl_Widget* ewl_htoolbar_new (void)`

Allocate a new toolbar widget with horizontal orientation.

Returns:

Returns pointer to new toolbar widget on success, NULL on failure.

6.73.4.2 `void ewl_toolbar_cb_child_add (Ewl_Container * c, Ewl_Widget * w)`

6.73.4.3 `void ewl_toolbar_icon_part_hide (Ewl_Toolbar * t, Ewl_Icon_Part part)`

Set the orientation of the toolbar.

Parameters:

t,: The toolbar to set the orientation on

part,: The **Ewl_Icon_Part** to hide

Returns:

Returns no value

6.73.4.4 int ewl_toolbar_init (Ewl_Toolbar * t)

Initialize the toolbar and inherited fields.

Parameters:

t,: the toolbar to initialize

Returns:

Returns TRUE on success, FALSE on failure.

Clears the contents of the toolbar and stores the default values.

6.73.4.5 Ewl_Widget* ewl_toolbar_new (void)

Allocate a new toolbar widget with default (horizontal) orientation.

Returns:

Returns pointer to new toolbar widget on success, NULL on failure.

6.73.4.6 Ewl_Orientation ewl_toolbar_orientation_get (Ewl_Toolbar * t)

This retrieves the current orientation of the toolbar.

Parameters:

t,: The toolbar to get the orientation from

Returns:

Returns the orientation currently set on the toolbar

6.73.4.7 void ewl_toolbar_orientation_set (Ewl_Toolbar * t, Ewl_Orientation o)

Set the orientation of the toolbar.

Parameters:

t,: The toolbar to set the orientation on

o,: The orientation to set on the toolbar

Returns:

Returns no value

6.73.4.8 Ewl_Widget* ewl_vtoolbar_new (void)

Allocate a new toolbar widget with vertical orientation.

Returns:

Returns pointer to new toolbar widget on success, NULL on failure.

6.74 Ewl_Tree: A Widget for List or Tree Layout

6.74.1 Detailed Description

Defines a widget for laying out other widgets in a tree or list like manner.

Remarks:

Inherits from **Ewl_Container** (p. 618).

Data Structures

- struct **Ewl_Tree**
*Inherits from **Ewl_Container** (p. 618) and extends to provide a tree layout.*
- struct **Ewl_Tree_Node**
*Inherits from **Ewl_Container** (p. 618) and extends to hold information on a row of the tree.*

Defines

- `#define EWL_TREE(t) ((Ewl_Tree *)t)`
- `#define EWL_TREE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE_TYPE))`
- `#define EWL_TREE_NODE(t) ((Ewl_Tree_Node *)t)`
- `#define EWL_TREE_NODE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE_NODE_TYPE))`
- `#define EWL_TREE_NODE_TYPE "node"`
- `#define EWL_TREE_TYPE "tree"`

Typedefs

- `typedef Ewl_Tree Ewl_Tree`
- `typedef Ewl_Tree_Node Ewl_Tree_Node`

Functions

- `void ewl_tree_cb_child_resize (Ewl_Container *c)`
- `void ewl_tree_cb_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_tree_cb_destroy (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_tree_cb_header_change (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_tree_cb_header_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_tree_cb_hscroll (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_tree_cb_node_child_add (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_tree_cb_node_child_del (Ewl_Container *c, Ewl_Widget *w, int idx)`
- `void ewl_tree_cb_node_child_hide (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_tree_cb_node_child_show (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_tree_cb_node_configure (Ewl_Widget *w, void *ev_data, void *user_data)`

- void **ewl_tree_cb_node_destroy** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_tree_cb_node_resize** (**Ewl_Container** *c, **Ewl_Widget** *w, int size, **Ewl_Orientation** o)
- void **ewl_tree_cb_node_toggle** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_tree_cb_row_hide** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_tree_cb_row_select** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_tree_columns_set** (**Ewl_Tree** *tree, unsigned short columns)

Change the number of columns displayed in a tree.

- **Ewl_Widget** * **ewl_tree_entry_row_add** (**Ewl_Tree** *tree, **Ewl_Row** *prow, char **text)

Add a row of text entries to a tree.

- unsigned int **ewl_tree_expandable_rows_get** (**Ewl_Tree** *tree)

Gets the header visibility of the tree.

- void **ewl_tree_expandable_rows_set** (**Ewl_Tree** *tree, unsigned int visible)

Sets the expandability of the rows in the tree.

- void **ewl_tree_headers_set** (**Ewl_Tree** *tree, char **headers)

Change the widgets in a trees column headers.

- unsigned int **ewl_tree_headers_visible_get** (**Ewl_Tree** *tree)

Gets the header visibility of the tree.

- void **ewl_tree_headers_visible_set** (**Ewl_Tree** *tree, unsigned int visible)

Sets the visibility of the headers in the tree.

- int **ewl_tree_init** (**Ewl_Tree** *tree, unsigned short columns)

Initialize the contents of a tree widget.

- **Ewl_Selection_Mode** **ewl_tree_mode_get** (**Ewl_Tree** *tree)

Retrieve the current selection mode of a tree.

- void **ewl_tree_mode_set** (**Ewl_Tree** *tree, **Ewl_Selection_Mode** mode)

Change the selection mode for a specified tree.

- **Ewl_Widget** * **ewl_tree_new** (unsigned short columns)

Allocate and initialize a new tree widget.

- void **ewl_tree_node_collapse** (**Ewl_Tree_Node** *tree)

Collapse a node in the tree.

- void **ewl_tree_node_expand** (**Ewl_Tree_Node** *tree)

Expand a node in the tree.

- int **ewl_tree_node_expandable_get** (**Ewl_Tree_Node** *node)

Retrieves the expand setting from the node.

- void **ewl_tree_node_expandable_set** (**Ewl_Tree_Node** *node, int expand)

Sets the expand value into the node.

- **int ewl_tree_node_init** (Ewl_Tree_Node *tree_node)
Initialize the node fields of an inheriting object.
- **Ewl_Widget * ewl_tree_node_new** (void)
Allocate and initialize a new node.
- **Ewl_Widget * ewl_tree_row_add** (Ewl_Tree *tree, Ewl_Row *prow, Ewl_Widget **children)
Add a group of widgets to a row in the tree.
- **Ewl_Widget * ewl_tree_row_column_get** (Ewl_Row *row, int i)
Retrieves the actual widget added via row_add instead of the cell.
- **void ewl_tree_row_destroy** (Ewl_Tree *tree, Ewl_Row *row)
Destroy a specified row from the tree.
- **void ewl_tree_row_expand_set** (Ewl_Row *row, Ewl_Tree_Node_Flags expanded)
Set the expand state of a specific row.
- **Ewl_Widget * ewl_tree_row_find** (Ewl_Tree *tree, int row)
- **void ewl_tree_selected_clear** (Ewl_Tree *tree)
Clear the current selection from a tree.
- **Ecore_List * ewl_tree_selected_get** (Ewl_Tree *tree)
Retrieves a list of selected rows from a tree.
- **Ewl_Widget * ewl_tree_text_row_add** (Ewl_Tree *tree, Ewl_Row *prow, char **text)
Add a row of text to a tree.

6.74.2 Define Documentation

6.74.2.1 #define EWL_TREE(t) ((Ewl_Tree *)t)

Typecasts a pointer to an **Ewl_Tree** (p. 769) pointer.

6.74.2.2 #define EWL_TREE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE_TYPE))

Returns TRUE if the widget is an **Ewl_Tree** (p. 769), FALSE otherwise

6.74.2.3 #define EWL_TREE_NODE(t) ((Ewl_Tree_Node *)t)

Typecasts a pointer to an **Ewl_Tree_Node** (p. 781) pointer.

6.74.2.4 `#define EWL_TREE_NODE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE_NODE_TYPE))`

Returns TRUE if the widget is an **Ewl_Tree_Node** (p. 781), FALSE otherwise

6.74.2.5 `#define EWL_TREE_NODE_TYPE "node"`

The type name for the **Ewl_Tree_Node** (p. 781) widget

6.74.2.6 `#define EWL_TREE_TYPE "tree"`

The type name for the **Ewl_Tree** (p. 769) widget

6.74.3 Typedef Documentation

6.74.3.1 `typedef struct Ewl_Tree Ewl_Tree`

The **Ewl_Tree** (p. 769) widget

6.74.3.2 `typedef struct Ewl_Tree_Node Ewl_Tree_Node`

A node in the tree

6.74.4 Function Documentation

- 6.74.4.1 void ewl_tree_cb_child_resize (Ewl_Container * *c*)
- 6.74.4.2 void ewl_tree_cb_configure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.74.4.3 void ewl_tree_cb_destroy (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.74.4.4 void ewl_tree_cb_header_change (Ewl_Widget * *w*, void * *ev*, void * *data*)
- 6.74.4.5 void ewl_tree_cb_header_configure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.74.4.6 void ewl_tree_cb_hscroll (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.74.4.7 void ewl_tree_cb_node_child_add (Ewl_Container * *c*, Ewl_Widget * *w*)
- 6.74.4.8 void ewl_tree_cb_node_child_del (Ewl_Container * *c*, Ewl_Widget * *w*, int *idx*)
- 6.74.4.9 void ewl_tree_cb_node_child_hide (Ewl_Container * *c*, Ewl_Widget * *w*)
- 6.74.4.10 void ewl_tree_cb_node_child_show (Ewl_Container * *c*, Ewl_Widget * *w*)
- 6.74.4.11 void ewl_tree_cb_node_configure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.74.4.12 void ewl_tree_cb_node_destroy (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.74.4.13 void ewl_tree_cb_node_resize (Ewl_Container * *c*, Ewl_Widget * *w*, int *size*, Ewl_Orientation *o*)
- 6.74.4.14 void ewl_tree_cb_node_toggle (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.74.4.15 void ewl_tree_cb_row_hide (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.74.4.16 void ewl_tree_cb_row_select (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.74.4.17 void ewl_tree_columns_set (Ewl_Tree * *tree*, unsigned short *columns*)

Change the number of columns displayed in a tree.

Parameters:

tree,: the tree to change the number of columns
columns,: the new number of columns to be displayed

Returns:

Returns no value.

The number of columns displayed in *tree* is changed to *columns*. When rows are added, pre-existing rows have empty contents in the additional columns which are appended. When rows are removed, previously existing rows destroy column contents that are removed from the end. If you need finer grain control over where columns are added or removed, see `ewl_tree_add_column` and `ewl_tree_del_column`.

6.74.4.18 `Ewl_Widget* ewl_tree_entry_row_add (Ewl_Tree * tree, Ewl_Row * prow, char ** text)`

Add a row of text entries to a tree.

Parameters:

tree,: the tree to hold the new entry row
*pro*w,: the parent row of the new entry row
text,: the array of strings that hold the entry text to be added

Returns:

Returns a pointer to a new row on success, NULL on failure.

6.74.4.19 `unsigned int ewl_tree_expandable_rows_get (Ewl_Tree * tree)`

Gets the header visibility of the tree.

Parameters:

tree,: The tree the get the header visibility

Returns:

Returns the header visibility of the tree

6.74.4.20 `void ewl_tree_expandable_rows_set (Ewl_Tree * tree, unsigned int expand)`

Sets the expandability of the rows in the tree.

Parameters:

tree,: The tree to set the row expandability
expand,: The expandability to set

Returns:

Returns no value.

6.74.4.21 void ewl_tree_headers_set (Ewl_Tree * *tree*, char ** *headers*)

Change the widgets in a trees column headers.

Parameters:

tree,: the tree to change column headers

headers,: the array of widget pointers containing the new headers

Returns:

Returns no value.

Stores the widgets in *headers* to header row of *tree*.

6.74.4.22 unsigned int ewl_tree_headers_visible_get (Ewl_Tree * *tree*)

Gets the header visibility of the tree.

Parameters:

tree,: The tree the get the header visibility

Returns:

Returns the header visibility of the tree

6.74.4.23 void ewl_tree_headers_visible_set (Ewl_Tree * *tree*, unsigned int *visible*)

Sets the visibility of the headers in the tree.

Parameters:

tree,: The tree to set the header visibility

visible,: The visibility to set

Returns:

Returns no value.

6.74.4.24 int ewl_tree_init (Ewl_Tree * *tree*, unsigned short *columns*)

Initialize the contents of a tree widget.

Parameters:

tree,: the tree widget to be initialized

columns,: the number of columns in the tree

Returns:

Returns TRUE on success, FALSE on failure.

The contents of the tree widget *tree* are initialized to their defaults, and the number of columns to display is set to *columns*.

6.74.4.25 Ewl_Selection_Mode ewl_tree_mode_get (Ewl_Tree * *tree*)

Retrieve the current selection mode of a tree.

Parameters:

tree,: the tree to get the selection mode

Returns:

Returns the current selection mode of the tree.

6.74.4.26 void ewl_tree_mode_set (Ewl_Tree * *tree*, Ewl_Selection_Mode *mode*)

Change the selection mode for a specified tree.

Parameters:

tree,: the tree to set the selection mode

mode,: the new selection mode for the tree

Returns:

Returns no value.

6.74.4.27 Ewl_Widget* ewl_tree_new (unsigned short *columns*)

Allocate and initialize a new tree widget.

Parameters:

columns,: the number of columns to display

Returns:

Returns NULL on failure, a new tree widget on success.

The paramater *columns* can be modified at a later time to display a different number of columns.

6.74.4.28 void ewl_tree_node_collapse (Ewl_Tree_Node * *node*)

Collapse a node in the tree.

Parameters:

node,: the node in the tree to collapse

Returns:

Returns no value. Hides the rows below *node*.

6.74.4.29 void ewl_tree_node_expand (Ewl_Tree_Node * *node*)

Expand a node in the tree.

Parameters:

node,: the node in the tree to expand

Returns:

Returns no value. Hides the rows below *node*.

6.74.4.30 int ewl_tree_node_expandable_get (Ewl_Tree_Node * *node*)

Retrieves the expand setting from the node.

Parameters:

node,: The node to work with

Returns:

Returns the expand setting of the node

6.74.4.31 void ewl_tree_node_expandable_set (Ewl_Tree_Node * *node*, int *expand*)

Sets the expand value into the node.

Parameters:

node,: The node to work with

expand,: The expand setting for the node

Returns:

Returns no value

6.74.4.32 int ewl_tree_node_init (Ewl_Tree_Node * *node*)

Initialize the node fields of an inheriting object.

Parameters:

node,: the node object to initialize

Returns:

Returns TRUE on success, FALSE on failure.

The fields of the *node* object are initialized to their defaults.

6.74.4.33 Ewl_Widget* ewl_tree_node_new (void)

Allocate and initialize a new node.

Returns:

Returns a newly allocated node on success, NULL on failure.

6.74.4.34 Ewl_Widget* ewl_tree_row_add (Ewl_Tree * *tree*, Ewl_Row * *prow*, Ewl_Widget ** *children*)

Add a group of widgets to a row in the tree.

Parameters:

tree,: the tree to hold the widgets

prow,: the parent row of the new row for the added widgets

children,: a NULL terminated array of widgets to add to the tree

Returns:

Returns a pointer to a new row on success, NULL on failure.

Adds a row to a specified *tree* with a parent row of *prow* and built from the widgets in the array *children*. The created row is nested below *prow*, and if *prow* is NULL the row is appended to the end of the list at the top level. The array *children* must be equal in size to the number of columns in *tree*. It is valid for the entries in *children* to be NULL, this creates an empty cell.

6.74.4.35 Ewl_Widget* ewl_tree_row_column_get (Ewl_Row * *row*, int *i*)

Retreives the actual widget added via row_add instead of the cell.

Parameters:

row,: the row to retrieve a column from

i,: the column to retrieve from the row

Returns:

Returns the widget in the column of the row sent

The behavior of this function is undefined if columns are not accounted for in the children of the row (if NULLs were passed in from row_add).

6.74.4.36 void ewl_tree_row_destroy (Ewl_Tree * *tree*, Ewl_Row * *row*)

Destroy a specified row from the tree.

Parameters:

tree,: the tree to destroy a row from

row,: the row to be destroyed from the tree

Returns:

Returns no value.

Removes *row* from *tree* if it is present in *tree*. The widgets in the row will be destroyed, so they should not be accessed at a later time.

6.74.4.37 `void ewl_tree_row_expand_set (Ewl_Row * row,
Ewl_Tree_Node_Flags expanded)`

Set the expand state of a specific row.

Parameters:

row,: the row to change the expanded state
expanded,: the new expanded state for the row

Returns:

Returns no value.

Changes the expanded state of *row* to *expanded*, which should be TRUE or FALSE.

6.74.4.38 `Ewl_Widget* ewl_tree_row_find (Ewl_Tree * tree, int row)`

Parameters:

tree,: The tree to find the row in
row,: The row number to find

Returns:

Returns the given row, or NULL if not found

6.74.4.39 `void ewl_tree_selected_clear (Ewl_Tree * tree)`

Clear the current selection from a tree.

Parameters:

tree,: the tree to clear the current selection

Returns:

Returns no value.

6.74.4.40 `Ecore_List* ewl_tree_selected_get (Ewl_Tree * tree)`

Retrieves a list of selected rows from a tree.

Parameters:

tree,: the tree to retrieve selected rows

Returns:

Returns a list of selected rows on success, NULL on failure.

6.74.4.41 `Ewl_Widget* ewl_tree_text_row_add (Ewl_Tree * tree, Ewl_Row * prow, char ** text)`

Add a row of text to a tree.

Parameters:

tree,: the tree to hold the new text row

prow,: the parent row of the new text row

text,: the array of strings that hold the text to be added

Returns:

Returns a pointer to a new row on success, NULL on failure.

6.75 Ewl_Tree2: A Widget for List or Tree Layout

6.75.1 Detailed Description

Defines a widget for laying out other widgets in a tree or list like manner. This also fulfills the functionality often seen in a table widget.

Model: Defines communication callbacks for views and controllers. Query row/column data, indicate expansion points, notify views and controllers of changes, trigger sorting on a row/column combination.

Data: Provides a wrapper to the data which allows for observer registration and a reference to a model for data access.

View: Defines the callbacks for setting up the widgets based on the data returned from the Model. Create widgets, set data on widgets, calculate sizing, minimize number of widgets.

Controller: The tree defines the entire controller and registers with the model. Provides controls to sort columns, expand/collapse branching points, reacts to changes from data model.

Simple case: Widgets packed using standard container functions. This should result in a simple list of packed widgets.

Features: * Default data provider, so standard container functions are supported. * MVC pattern of data access for more powerful layouts. * Packing of arbitrary widget types in cells. * Theming based on repeating row counts. * Number of columns dependant on number of providers, not a fixed number. * When using data providers, can reap widgets to save memory and decrease layout time.

Issues: * How do we support click callbacks on full rows? Special notifier necessary? Proposal: Tree value change callback on row selection. Passes event structure with array of selected row numbers. Simple case handled by callback on packed widgets. * Column or row layout? Column allows for redirecting to a container to handle the case of container functions used to add widgets and to use a fixed number of rows in display region. It would also allow for skipping configure of an entire off-screen column, rows would lay out the cells off-screen. How do we implement column layout? Proposal: Row height cache, easy for fixed height rows, more complex for variable height.

Remarks:

Inherits from **Ewl_MVC** (p. 705).

6.75.2 Tutorial

Finding the Tree in the Forest (originally at <http://everburning.com/news/finding-the-tree-in-the-forest>) We've been doing a bunch of work on **Ewl_Tree2** (p. 771) lately. Its been shaping up nicely at the moment. To that end, I thought I'd do a quick write up on how it works and what you can do with it.

Ewl_Tree2 (p. 771) is built based on an MVC (model/view/controller) framework. This makes it a lot easier for developers to keep their data up to date without having to go through all kinds of contortions using the tree nodes and widgets as they do with the current **Ewl_Tree** (p. 769) widget.

There are three basic items you'll need to become familiar with in order to use **Ewl_Tree2** (p. 771). They are, **Ewl_Tree2** (p. 771), of course, **Ewl_Model** (p. 703) and **Ewl_View** (p. 784). These three will allow you to setup your columns and the tree.

The best way to show something is through an example, so that's what I'll do. This is basically

the tree2 test case from ewl_test ported to run in a window. The case is pretty simple. We store an array of data, this array contains nodes that specify the text and an image to be displayed.

We then create a three column tree. The first column will show the text as an **Ewl_Label** (p.692) widget. The second column will display an **Ewl_Image** (p.686). The third will be an **Ewl_Button** (p.589). The third column doesn't use the ewl_button code directly as we want to set two pieces of information so we write our own methods to handle the creating and assignment functions.

I'll be putting the code into tree2_test.c and using the following to compile the code as I go.

```
oni:~/dev/tree2_test$ gcc -o tree2_test tree2_test.c `ewl-config --cflags --libs`
```

With that out of the way, on with the show. Im going to start by listing all of the code and then Ill go through it piece by piece.

```
#include <Ewl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define DATA_ELEMENTS 5

typedef struct Test_Row_Data Test_Row_Data;
struct Test_Row_Data
{
    char *image;
    char *text;
};

typedef struct Test_Data Test_Data;
struct Test_Data
{
    unsigned int count;
    Test_Row_Data **rows;
};

static void *test_data_setup(void);

static Ewl_Widget *test_custom_new(void);
static void test_custom_assign_set(Ewl_Widget *w, void *data);

static Ewl_Widget *test_data_header_fetch(void *data, int column);
static void *test_data_fetch(void *data, unsigned int row, unsigned int column);
static void test_data_sort(void *data, unsigned int column, Ewl_Sort_Direction sort);
static int test_data_count_get(void *data);

static void cb_delete_window(Ewl_Widget *w, void *ev, void *data);
static void cb_scroll_headers(Ewl_Widget *w, void *ev, void *data);
static void cb_scroll_visible(Ewl_Widget *w, void *ev, void *data);

int
main(int argc, char ** argv)
{
    Ewl_Widget *tree, *box, *o, *o2;
    Ewl_Model *model;
    Ewl_View *view;
    void *data;

    // make sure we can setup ewl
    if (!ewl_init(&argc, argv))
    {
        fprintf(stderr, "Unable to init ewl.n");
        return 1;
    }
}
```

```

// create the window
o = ewl_window_new();
ewl_window_title_set(EWL_WINDOW(o), "tree2 example");
ewl_window_class_set(EWL_WINDOW(o), "tree2_example");
ewl_window_name_set(EWL_WINDOW(o), "tree2_example");
ewl_object_size_request(EWL_OBJECT(o), 640, 480);
ewl_callback_append(o, EWL_CALLBACK_DELETE_WINDOW, cb_delete_window, NULL);
ewl_widget_show(o);

box = ewl_vbox_new();
ewl_container_child_append(EWL_CONTAINER(o), box);
ewl_widget_show(box);

o2 = ewl_hbox_new();
ewl_container_child_append(EWL_CONTAINER(box), o2);
ewl_object_fill_policy_set(EWL_OBJECT(o2),
    EWL_FLAG_FILL_VSHRINK | EWL_FLAG_FILL_HFILL);
ewl_widget_show(o2);

// create our data
data = test_data_setup();

// create the model that'll be used for the first two columns
model = ewl_model_new();
ewl_model_data_fetch_set(model, test_data_fetch);
ewl_model_data_sort_set(model, test_data_sort);
ewl_model_data_count_set(model, test_data_count_get);

tree = ewl_tree2_new();
ewl_container_child_append(EWL_CONTAINER(box), tree);
ewl_object_fill_policy_set(EWL_OBJECT(tree), EWL_FLAG_FILL_ALL);
ewl_tree2_data_set(EWL_TREE2(tree), data);
ewl_widget_show(tree);

// create a view for the first column that just has an ewl label
view = ewl_view_new();
ewl_view_constructor_set(view, ewl_label_new);
ewl_view_assign_set(view, EWL_VIEW_ASSIGN(ewl_label_text_set));
ewl_view_header_fetch_set(view, test_data_header_fetch);
ewl_tree2_column_append(EWL_TREE2(tree), model, view);

// create a view for the second column that just has an ewl image
view = ewl_view_new();
ewl_view_constructor_set(view, ewl_image_new);
ewl_view_assign_set(view, EWL_VIEW_ASSIGN(ewl_image_file_path_set));
ewl_view_header_fetch_set(view, test_data_header_fetch);
ewl_tree2_column_append(EWL_TREE2(tree), model, view);

// we don't want this one sortable
model = ewl_model_new();
ewl_model_data_fetch_set(model, test_data_fetch);
ewl_model_data_count_set(model, test_data_count_get);

// create a view for the third column that has a custom widget
view = ewl_view_new();
ewl_view_constructor_set(view, test_custom_new);
ewl_view_assign_set(view, test_custom_assign_set);
ewl_view_header_fetch_set(view, test_data_header_fetch);
ewl_tree2_column_append(EWL_TREE2(tree), model, view);

// create the checkbuttons for the top box
o = ewl_checkbutton_new();
ewl_button_label_set(EWL_BUTTON(o), "Scroll headers");
ewl_container_child_append(EWL_CONTAINER(o2), o);
ewl_callback_append(o, EWL_CALLBACK_CLICKED,
    cb_scroll_headers, tree);

```

```

    ewl_widget_show(o);

    o = ewl_checkbutton_new();
    ewl_button_label_set(EWL_BUTTON(o), "Scroll visible");
    ewl_container_child_append(EWL_CONTAINER(o2), o);
    ewl_checkbutton_checked_set(EWL_CHECKBUTTON(o), TRUE);
    ewl_callback_append(o, EWL_CALLBACK_CLICKED,
        cb_scroll_visible, tree);
    ewl_widget_show(o);

    ewl_main();
    return 0;
}

// setup our data
static void *
test_data_setup(void)
{
    Test_Data *data;
    Test_Row_Data **dt;

    data = calloc(1, sizeof(Test_Data));
    dt = calloc(DATA_ELEMENTS, sizeof(Test_Row_Data *));

    dt[0] = calloc(1, sizeof(Test_Row_Data));
    dt[0]->image = strdup("/usr/local/share/ewl/images/e-logo.png");
    dt[0]->text = strdup("The E logo");

    dt[1] = calloc(1, sizeof(Test_Row_Data));
    dt[1]->image = strdup("/usr/local/share/ewl/images/elicit.png");
    dt[1]->text = strdup("The Elicit image");

    dt[2] = calloc(1, sizeof(Test_Row_Data));
    dt[2]->image = strdup("/usr/local/share/ewl/images/entrance.png");
    dt[2]->text = strdup("The Entrance image");

    dt[3] = calloc(1, sizeof(Test_Row_Data));
    dt[3]->image = strdup("/usr/local/share/ewl/images/End.png");
    dt[3]->text = strdup("Zebra");

    dt[4] = calloc(1, sizeof(Test_Row_Data));
    dt[4]->image = strdup("/usr/local/share/ewl/images/banner-top.png");
    dt[4]->text = strdup("Ant");

    data->rows = dt;
    data->count = DATA_ELEMENTS;

    return data;
}

static Ewl_Widget *
test_custom_new(void)
{
    Ewl_Widget *button;

    button = ewl_button_new();

    return button;
}

static void
test_custom_assign_set(Ewl_Widget *w, void *data)
{
    Test_Row_Data *d;

    d = data;
    ewl_button_label_set(EWL_BUTTON(w), d->text);

```

```

    ewl_button_image_set(EWL_BUTTON(w), d->image, NULL);
}

static Ewl_Widget *
test_data_header_fetch(void *data , int column)
{
    Ewl_Widget *l;

    l = ewl_label_new();
    if (column == 0)
        ewl_label_text_set(EWL_LABEL(l), "Title");
    else if (column == 1)
        ewl_label_text_set(EWL_LABEL(l), "Image");
    else
        ewl_label_text_set(EWL_LABEL(l), "Button");
    ewl_widget_show(l);

    return l;
}

static void *
test_data_fetch(void *data, unsigned int row, unsigned int column)
{
    Test_Data *d;
    void *val = NULL;

    d = data;

    if (column == 0)
        val = d->rows[row]->text;

    else if (column == 1)
        val = d->rows[row]->image;

    else if (column == 2)
        val = d->rows[row];

    return val;
}

static void
test_data_sort(void *data, unsigned int column, Ewl_Sort_Direction sort)
{
    Test_Data *d;
    int i;

    // just leave it if we're in sort none.
    if (sort == EWL_SORT_DIRECTION_NONE)
        return;

    d = data;

    for (i = (DATA_ELEMENTS - 1); i >= 0; i--)
    {
        int j;

        for (j = 1; j <= i; j++)
        {
            char *a, *b;

            if (column == 0)
            {
                a = d->rows[j - 1]->text;
                b = d->rows[j]->text;
            }
            else
            {

```

```

        a = d->rows[j - 1]->image;
        b = d->rows[j]->image;
    }

    if (((sort == EWL_SORT_DIRECTION_ASCENDING) && strcmp(a, b) > 0)
        || ((sort == EWL_SORT_DIRECTION_DESCENDING)
            && strcmp(a, b) < 0))
    {
        char *temp;

        temp = d->rows[j - 1]->text;
        d->rows[j - 1]->text = d->rows[j]->text;
        d->rows[j]->text = temp;

        temp = d->rows[j - 1]->image;
        d->rows[j - 1]->image = d->rows[j]->image;
        d->rows[j]->image = temp;
    }
}

static int
test_data_count_get(void *data)
{
    Test_Data *d;

    d = data;

    return d->count;
}

static void
cb_delete_window(Ewl_Widget *w, void *ev, void *data)
{
    ewl_widget_destroy(w);
    ewl_main_quit();
}

static void
cb_scroll_headers(Ewl_Widget *w, void *ev, void *data)
{
    Ewl_Tree2 *tree;

    tree = data;
    ewl_tree2_scroll_headers_set(tree,
        ewl_checkbutton_is_checked(EWL_CHECKBUTTON(w)));
}

static void
cb_scroll_visible(Ewl_Widget *w, void *ev, void *data)
{
    Ewl_Tree2 *tree;

    tree = data;
    ewl_tree2_scroll_visible_set(tree,
        ewl_checkbutton_is_checked(EWL_CHECKBUTTON(w)));
}

```

Simple enough, eh? Ok, maybe I should go through it then.

```

#include <Ewl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

```

We start off with the standard set of includes. **Ewl.h** (p.817) is obviously required to do any

EWL programming. The others are needed as well be using functions they define throughout the application.

```
#define DATA_ELEMENTS 5

typedef struct Test_Row_Data Test_Row_Data;
struct Test_Row_Data
{
    char *image;
    char *text;
};

typedef struct Test_Data Test_Data;
struct Test_Data
{
    unsigned int count;
    Test_Row_Data **rows;
};
```

I'm not planning on doing anything fancy with my data. Just keeping an array with five elements. Im using a define DATA_ELEMENTS to store the number of elements as Ill be using this in a few places. The data will be stored in a Test_Data structure. This struct will store the number of items in the array and an array of Test_Row_Data pointers. Test_Row_Data structs just store the text and image strings for each of our rows.

```
static void *test_data_setup(void);

static Ewl_Widget *test_custom_new(void);
static void test_custom_assign_set(Ewl_Widget *w, void *data);

static Ewl_Widget *test_data_header_fetch(void *data, int column);
static void *test_data_fetch(void *data, unsigned int row, unsigned int column);
static void test_data_sort(void *data, unsigned int column, Ewl_Sort_Direction sort);
static int test_data_count_get(void *data);

static void cb_delete_window(Ewl_Widget *w, void *ev, void *data);
static void cb_scroll_headers(Ewl_Widget *w, void *ev, void *data);
static void cb_scroll_visible(Ewl_Widget *w, void *ev, void *data);
```

As you can see, a bunch of pre-declarations next. Well be seeing, and getting the explanation for these as we go along.

```
int
main(int argc, char ** argv)
{
    Ewl_Widget *tree, *box, *o, *o2;
    Ewl_Model *model;
    Ewl_View *view;
    void *data;

    // make sure we can setup ewl
    if (!ewl_init(&argc, argv))
    {
        fprintf(stderr, "Unable to init ewl.n");
        return 1;
    }
}
```

The first step in any EWL application is to initialize EWL itself. This is done with a call to **ewl_init()** (p.313). **ewl_init()** (p.313) accepts two parameters, the argc and argv arguments that were passed to your application. **ewl_init()** (p.313) will return TRUE if EWL was successfully initialized or FALSE otherwise. The reason to pass the args to **ewl_init()** (p.313) is so that EWL

can parse out any EWL specific arguments. Things like setting the rendering engine or printing the EWL help documentation. These parameters can both be safely set to NULL if desired.

With EWL setup we can get down to the fun bit of creating the UI.

```
// create the window
o = ewl_window_new();
ewl_window_title_set(EWL_WINDOW(o), "tree2 example");
ewl_window_class_set(EWL_WINDOW(o), "tree2_example");
ewl_window_name_set(EWL_WINDOW(o), "tree2_example");
ewl_object_size_request(EWL_OBJECT(o), 640, 480);
ewl_callback_append(o, EWL_CALLBACK_DELETE_WINDOW, cb_delete_window, NULL);
ewl_widget_show(o);

box = ewl_vbox_new();
ewl_container_child_append(EWL_CONTAINER(o), box);
ewl_widget_show(box);

o2 = ewl_hbox_new();
ewl_container_child_append(EWL_CONTAINER(box), o2);
ewl_object_fill_policy_set(EWL_OBJECT(o2), EWL_FLAG_FILL_VSHRINK | EWL_FLAG_FILL_HFILL);
ewl_widget_show(o2);
```

This little chunks sets up our base UI. We create the **Ewl_Window** (p.788) first using **ewl_window_new()** (p.571). We set the title, class and name for the window and then give it a default size of 640x480. Once this is done we set a callback for when the window is destroyed. This is done with **ewl_callback_append()** (p.47) call. We want to get notified when the window receives the **EWL_CALLBACK_DELETE_WINDOW** callback by having the **cb_delete_window()** function executed.

With the main window setup we create an **Ewl_Box** (p.586) inside of it. An **Ewl_Window** (p.788) by default has no layout policy so if you pack several widgets into it theyll all be sitting on top of each other. You have to pack a box, or something, in there to handle the layout of the contents. Youll see we use **ewl_container_child_append()** (p.90) to add the box to the window. Youll be seeing this a lot as we pack the UI together. There are also **ewl_container_child_prepend()** (p.95) and **ewl_container_child_insert()** (p.93) calls that can be used.

We also create a second **Ewl_Box** (p.586) to hold some checkbuttons that well create later. Were creating the box here just because I like to keep things order. Its the first item in the window so I create and pack it first. We could also use **ewl_container_child_prepend()** (p.95) later to add it to the box if we wished. Were setting a custom fill policy on this box with **ewl_object_fill_policy_set()** (p.358). The policy were setting is **EWL_FLAG_FILL_VSHRINK | EWL_FLAG_FILL_HFILL**. Were telling the widget that we want it to be as small as possible vertically but take up as much space as possible horizontally. The fill policy is just a bit mask so were bit wise oring the values together.

I guess a quick note about EWL inheritance is necessary at some point. EWL uses an object oriented approach to its widgets. All widgets inherit from **Ewl_Widget** (p.785). **Ewl_Widget** (p.785) inherits from **Ewl_Object** (p.709). So any **ewl_widget_*** or **ewl_object_*** call can be called on any widgets. You just need to cast to the correct type first. The casts are done with the **EWL_WIDGET()** (p.545) or **EWL_OBJECT()** (p.348) macros. All widgets in EWL have an **EWL_WIDGET_NAME()** macro. Theres a lot of inheritance going on in EWL. The **Ewl_Box** (p.586) code inherits from the **Ewl_Container** (p.618) code. So any **ewl_container_*** call will work on an **Ewl_Box** (p.586), you just have to wrap the box variable with an **EWL_CONTAINER()** (p.87) call. Take a look at the EWL docs for a complete listing of the inheritance. (Or if youre looking at the header files the inheritance is always the first item in the struct.)

```
// create our data
data = test_data_setup();
```


This is just a convenience function to create our data array. I like to keep stuff separated out whenever possible. We'll see what this does later.

```
// create the model that'll be used for the first two columns
model = ewl_model_new();
ewl_model_data_fetch_set(model, test_data_fetch);
ewl_model_data_sort_set(model, test_data_sort);
ewl_model_data_count_set(model, test_data_count_get);
```

Our first two columns in the tree will actually use the same data model. I don't want the third column to be sortable so we need to use a slightly different model (although it's almost the same). We need to set at a minimum two pieces of data into the **Ewl_Model** (p. 703). These are, the function to fetch the model data, set with **ewl_model_data_fetch_set()** (p. 322), and the function to get a count of the number of rows of data, set with **ewl_model_data_count_set()** (p. 321). I'm using the third **ewl_model_data_sort_set()** (p. 323) to set a sort function for the first two columns. Each of these calls takes the model and a function pointer. You can see the function signature at the top of the file. We'll take a closer look at these functions a bit later.

```
tree = ewl_tree2_new();
ewl_container_child_append(EWL_CONTAINER(box), tree);
ewl_object_fill_policy_set(EWL_OBJECT(tree), EWL_FLAG_FILL_ALL);
ewl_tree2_data_set(EWL_TREE2(tree), data);
ewl_widget_show(tree);
```

Next we create the tree itself. The tree is packed into the box we created earlier. We set the box with a fill policy of **EWL_FLAG_FILL_ALL** so it will take as much, or as little space as necessary. We then set the data we created into the tree with **ewl_tree2_data_set()** function. This data will be passed around to our various functions as the tree does its work.

```
// create a view for the first column that just has an ewl label
view = ewl_view_new();
ewl_view_constructor_set(view, ewl_label_new);
ewl_view_assign_set(view, EWL_VIEW_ASSIGN(ewl_label_text_set));
ewl_view_header_fetch_set(view, test_data_header_fetch);
ewl_tree2_column_append(EWL_TREE2(tree), model, view);
```

Ok, with our tree created we can start to add columns. We've already created the **Ewl_Model** (p. 703) for the first two columns so we just need to create the views.

As I mentioned before the first column is using **Ewl_Label** (p. 692) to display the text of the data item. So, we create a new **Ewl_View** (p. 784) with **ewl_view_new()** (p. 537). We then set the constructor for this column with **ewl_view_constructor_set()**. This will be used to create a new widget for the item in the column. Then we use **ewl_view_assign_set()** to set the function that will be used to assign data into our widget for a given cell. Tree columns need a header. We'll set a third callback function to get the header data for the tree. We use **ewl_view_header_fetch_set()** (p. 537) to set this function.

Once our view is created we call **ewl_tree2_column_append()** to create a new column in the tree using our model and view.

```
// create a view for the second column that just has an ewl image view =
ewl_view_new();
ewl_view_constructor_set(view, ewl_image_new);
ewl_view_assign_set(view, EWL_VIEW_ASSIGN(ewl_image_file_path_set));
ewl_view_header_fetch_set(view, test_data_header_fetch);
ewl_tree2_column_append(EWL_TREE2(tree), model, view);
```

The second column in the tree is created the same as the first except we use the **Ewl_Image** (p.686) functions instead of the **Ewl_Label** (p.692) functions. You can use any widget within EWL to setup a view. This gives the tree the flexibility of being able to display any desired widget, even custom designed widgets.

```
// we don't want this one sortable
model = ewl_model_new();
ewl_model_data_fetch_set(model, test_data_fetch);
ewl_model_data_count_set(model, test_data_count_get);

// create a view for the third column that has a custom widget
view = ewl_view_new();
ewl_view_constructor_set(view, test_custom_new);
ewl_view_assign_set(view, test_custom_assign_set);
ewl_view_header_fetch_set(view, test_data_header_fetch);
ewl_tree2_column_append(EWL_TREE2(tree), model, view);
```

The third column in the tree is similar to the first two except we dont want it sortable. So, we create a new **Ewl_Model** (p.703) and set the fetch and count functions as before but skip the sort function. We then create the view and use our custom constructor and assignment functions instead of a standard EWL set. With the model and view created we append the column into the tree.

```
// create the checkbuttons for the top box
o = ewl_checkbutton_new();
ewl_button_label_set(EWL_BUTTON(o), "Scroll headers");
ewl_container_child_append(EWL_CONTAINER(o2), o);
ewl_callback_append(o, EWL_CALLBACK_CLICKED, cb_scroll_headers, tree);
ewl_widget_show(o);

o = ewl_checkbutton_new();
ewl_button_label_set(EWL_BUTTON(o), "Scroll visible");
ewl_container_child_append(EWL_CONTAINER(o2), o);
ewl_checkbutton_checked_set(EWL_CHECKBUTTON(o), TRUE);
ewl_callback_append(o, EWL_CALLBACK_CLICKED, cb_scroll_visible, tree);
ewl_widget_show(o);
```

The final piece of the UI to be added are the two checkbuttons. These will be used to change some settings of the tree so we can see some of the different options. The both respond to **EWL_CALLBACK_CLICKED** callbacks the first calling the `cb_scroll_headers()` function and the second calling `cb_scroll_visible()` function.

```
ewl_main();
return 0;
}
```

Finally we call **ewl_main()** (p.314) to kick off the main EWL event loop. When **ewl_main()** (p.314) is done then were finished so just return. **ewl_main_quit()** (p.314) will actually call **ewl_shutdown()** (p.314) for us so we dont have to worry about it.

Ok, with our UI out of the way all we need to deal with are all the functions weve referenced. First up is creating our data. For this example were just using an array but the thing to keep in mind is you can use anything to store your data. EWL never accesses this data directly it always does it through the calls you specified. So, if you want to use an **Evas_List** or an **Ecore_List** or a tree or some other structure its up to you. EWL doesnt care.

```
// setup our data
static void *
test_data_setup(void)
```

```

{
    Test_Data *data;
    Test_Row_Data **dt;

    data = calloc(1, sizeof(Test_Data));
    dt = calloc(DATA_ELEMENTS, sizeof(Test_Row_Data *));

    dt[0] = calloc(1, sizeof(Test_Row_Data));
    dt[0]->image = strdup("/usr/local/share/ewl/images/e-logo.png");
    dt[0]->text = strdup("The E logo");

    dt[1] = calloc(1, sizeof(Test_Row_Data));
    dt[1]->image = strdup("/usr/local/share/ewl/images/elicit.png");
    dt[1]->text = strdup("The Elicit image");

    dt[2] = calloc(1, sizeof(Test_Row_Data));
    dt[2]->image = strdup("/usr/local/share/ewl/images/entrance.png");
    dt[2]->text = strdup("The Entrance image");

    dt[3] = calloc(1, sizeof(Test_Row_Data));
    dt[3]->image = strdup("/usr/local/share/ewl/images/End.png");
    dt[3]->text = strdup("Zebra");

    dt[4] = calloc(1, sizeof(Test_Row_Data));
    dt[4]->image = strdup("/usr/local/share/ewl/images/banner-top.png");
    dt[4]->text = strdup("Ant");

    data->rows = dt;
    data->count = DATA_ELEMENTS;

    return data;
}

```

Nothing fancy in there so Im not going to bother explaining it.

```

static Ewl_Widget *
test_custom_new(void)
{
    Ewl_Widget *button;
    button = ewl_button_new();
    return button;
}

```

As I mentioned for our third column were using a custom constructor and assignment calls. The reason for this, since we just want a simple button, is that we want to set two pieces of data into the widget instead of just one. We could actually just use `ewl_button_new()` (p. 40) in the view and have a custom assignment function, but this makes for a better example. For the constructor all we do is create our widget and return it. Its as simple as that.

```

static void
test_custom_assign_set(Ewl_Widget *w, void *data)
{
    Test_Row_Data *d;

    d = data;
    ewl_button_label_set(EWL_BUTTON(w), d->text);
    ewl_button_image_set(EWL_BUTTON(w), d->image, NULL);
}

```

For the assignment part of the custom widget EWL will provide the widget created with the view, w, and the piece of data that the rows model returned for the current cell in the tree, data. Using these we can setup the widget however we see fit. In this case were setting the label and image of the button.

```
static Ewl_Widget *
test_data_header_fetch(void *data , int column)
{
    Ewl_Widget *l;

    l = ewl_label_new();
    if (column == 0)
        ewl_label_text_set(EWL_LABEL(l), "Title");
    else if (column == 1)
        ewl_label_text_set(EWL_LABEL(l), "Image");
    else
        ewl_label_text_set(EWL_LABEL(l), "Button");

    ewl_widget_show(l);

    return l;
}
```

The last view function we need to worry about is `test_data_header_fetch()`. EWL will call this function for each column to get the widget to display in the header. The data pointer is the data set on the tree itself and the column is the column number to return the header for (column numbers start at 0).

In this case were just creating an **Ewl_Label** (p. 692) and setting and appropriate bit of text to describe the column.

Not too bad so far, right?

With the view code out of the way lets move on to model code. We start with the fetch function. This is called whenever EWL needs to know what information to pass to the views assign function.

```
static void *
test_data_fetch(void *data, unsigned int row, unsigned int column)
{
    Test_Data *d;
    void *val = NULL;

    d = data;

    if (column == 0)
        val = d->rows[row]->text;
    else if (column == 1)
        val = d->rows[row]->image;
    else if (column == 2)
        val = d->rows[row];

    return val;
}
```

The fetch function returns a `void *` as EWL is making no assumptions about what type of data youll need to pass into your assign function. In this case we return a `char *` for the first two columns and a `Tree_Row_Data *` struct for the third column. The fetch function is passed the data set into the tree, the row and the column that we are interested in.

If youve set a sort function into your model then the tree headers become clickable. When the user clicks a header the sort function is called for the given column. With this app Im just using a simple bubble sort. Youll possibly want to use something a bit better for a real application.

```
static void
test_data_sort(void *data, unsigned int column, Ewl_Sort_Direction sort)
{
    Test_Data *d;
    int i;
```

```

// just leave it if we're in sort none.
if (sort == EWL_SORT_DIRECTION_NONE)
    return;

d = data;

for (i = (DATA_ELEMENTS - 1); i >= 0; i--)
{
    int j;

    for (j = 1; j <= i; j++)
    {
        char *a, *b;

        if (column == 0)
        {
            a = d->rows[j - 1]->text;
            b = d->rows[j]->text;
        }
        else
        {
            a = d->rows[j - 1]->image;
            b = d->rows[j]->image;
        }

        if (((sort == EWL_SORT_DIRECTION_ASCENDING) && strcmp(a, b) > 0)
            || ((sort == EWL_SORT_DIRECTION_DESCENDING)
                && strcmp(a, b) < 0))
        {
            char *temp;

            temp = d->rows[j - 1]->text;
            d->rows[j - 1]->text = d->rows[j]->text;
            d->rows[j]->text = temp;

            temp = d->rows[j - 1]->image;
            d->rows[j - 1]->image = d->rows[j]->image;
            d->rows[j]->image = temp;
        }
    }
}
}

```

The sort call has three parameters. The the data we set on the tree, the column number that we are sorting and the direction of the sort. The three possible sort directions are `EWL_SORT_DIRECTION_NONE`, `EWL_SORT_DIRECTION_ASCENDING` and `EWL_SORT_DIRECTION_DESCENDING`. In this case, we dont bother doing anything if we are set to a sort of `EWL_SORT_DIRECTION_NONE`. We then sort the data as needed in either ascending or descending order based on the sort parameter.

```

static int
test_data_count_get(void *data)
{
    Test_Data *d;

    d = data;

    return d->count;
}

```

The last model function we need to implement is the count function. This is just a way for your application to tell EWL how many rows are in your data. In this case we just return the count parameter.

Thats it for the model code. All thats left are the three callbacks we defined for the UI.

```
static void
cb_delete_window(Ewl_Widget *w, void *ev, void *data)
{
    ewl_widget_destroy(w);
    ewl_main_quit();
}
```

`cb_delete_window()` will be called when the window is deleted. All we do is destroy the window with a call to **ewl_widget_destroy()** (p.552) and quit the application with **ewl_main_quit()** (p.314). Because we attached this callback to the **Ewl_Window** (p.788) widget the first parameter `*w` is our window. You dont technically need to **ewl_widget_destroy()** (p.552) the window but its nice to clean up after yourself.

```
static void
cb_scroll_headers(Ewl_Widget *w, void *ev , void *data)
{
    Ewl_Tree2 *tree;

    tree = data;
    ewl_tree2_scroll_headers_set(tree, ewl_checkbutton_is_checked(EWL_CHECKBUTTON(w)));
}
```

The `cb_scroll_headers()` function toggles if the headers in the tree should be scrolled. This is done by calling `ewl_tree2_scroll_headers_set()` and passing either `TRUE` or `FALSE` depending on if we want the headers scrolled. This is conveniently the same as the return of **ewl_checkbutton_is_checked()** (p.58).

```
static void
cb_scroll_visible(Ewl_Widget *w, void *ev , void *data)
{
    Ewl_Tree2 *tree;

    tree = data;
    ewl_tree2_scroll_visible_set(tree, ewl_checkbutton_is_checked(EWL_CHECKBUTTON(w)));
}
```

The `cb_scroll_visible()` function works in the same way as the `cb_scroll_headers()` function except that it either enables or disables the display of the scrollbars in the tree.

With that, were at the end of our code. If you use the compilation line given above everything should work out and when you run the app you should seeing something similar too:

Hopefully that wasnt too painful and you can see how easily it is to work with the new **Ewl_Tree2** (p.771) code.

Oh, before I forget. If you're updating your model and you want EWL to redraw the tree you just need to call `ewl_tree2_dirty_set()` and pass `TRUE` as the second parameter. This will signal EWL that something has changed in the model and the tree will be redrawn.

Data Structures

- struct **Ewl_Tree2**

*Inherits from **Ewl_MVC** (p.705) and extends to provide a tree widget.*

- struct **Ewl_Tree2_Node**

*Inherits from **Ewl_Container** (p.618) and extends to hold information on a row of the tree.*

Defines

- `#define EWL_TREE2(t) ((Ewl_Tree2 *)t)`
- `#define EWL_TREE2_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE2_TYPE))`
- `#define EWL_TREE2_NODE(n) ((Ewl_Tree2_Node *)n)`
- `#define EWL_TREE2_NODE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE2_NODE_TYPE))`
- `#define EWL_TREE2_NODE_TYPE "node"`
- `#define EWL_TREE2_TYPE "tree2"`

Typedefs

- `typedef Ewl_Tree2 Ewl_Tree2`
- `typedef Ewl_Tree2_Node Ewl_Tree2_Node`

Functions

- `void ewl_tree2_cb_column_sort (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_tree2_cb_configure (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_tree2_cb_destroy (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_tree2_cb_node_child_add (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_tree2_cb_node_child_del (Ewl_Container *c, Ewl_Widget *w, int idx)`
- `void ewl_tree2_cb_node_child_hide (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_tree2_cb_node_child_show (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_tree2_cb_node_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_tree2_cb_node_realize (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_tree2_cb_node_resize (Ewl_Container *c, Ewl_Widget *w, int size, Ewl_Orientation o)`
- `void ewl_tree2_cb_node_toggle (Ewl_Widget *w, void *ev_data, void *user_data)`
- `unsigned int ewl_tree2_column_count_get (Ewl_Tree2 *tree)`
Retrives the number of columns in the tree.
- `void ewl_tree2_column_count_set (Ewl_Tree2 *tree, unsigned int count)`
Sets the number of columns in the tree.
- `Ewl_View * ewl_tree2_content_view_get (Ewl_Tree2 *tree)`
Retrives the view used to generate the tree content area.
- `void ewl_tree2_content_view_set (Ewl_Tree2 *tree, Ewl_View *view)`
Sets the view to use to generate the content area.
- `Ewl_Widget * ewl_tree2_content_widget_get (Ewl_Tree2 *tree)`
Retrieves the widget containing the tree rows.
- `unsigned int ewl_tree2_fixed_rows_get (Ewl_Tree2 *tree)`
Retrieve the fixed row size of the tree.
- `void ewl_tree2_fixed_rows_set (Ewl_Tree2 *tree, unsigned int fixed)`

Set the fixed row size of the tree.

- unsigned int **ewl_tree2_headers_visible_get** (**Ewl_Tree2** *tree)
Retrieve if the header is visible in the tree.
- void **ewl_tree2_headers_visible_set** (**Ewl_Tree2** *tree, unsigned char visible)
Toggle if the header is visible in the tree.
- int **ewl_tree2_init** (**Ewl_Tree2** *tree)
Initialize the contents of a tree widget.
- **Ewl_Widget** * **ewl_tree2_new** (void)
Allocate and initialize a new tree widget.
- void **ewl_tree2_node_collapse** (**Ewl_Tree2_Node** *node)
- void **ewl_tree2_node_expand** (**Ewl_Tree2_Node** *node)
- unsigned int **ewl_tree2_node_expandable_get** (**Ewl_Tree2_Node** *node)
- void **ewl_tree2_node_expandable_set** (**Ewl_Tree2_Node** *node, void *data)
- unsigned int **ewl_tree2_node_expanded_is** (**Ewl_Tree2_Node** *node)
- int **ewl_tree2_node_init** (**Ewl_Tree2_Node** *node)
- **Ewl_Widget** * **ewl_tree2_node_new** (void)
- void **ewl_tree2_row_collapse** (**Ewl_Tree2** *tree, void *data, unsigned int row)
Sets the given row to collapsed for the given data in tree.
- void **ewl_tree2_row_expand** (**Ewl_Tree2** *tree, void *data, unsigned int row)
When the tree displays the data in data it will expand the give row. This data is the parent of the expansion row.
- unsigned int **ewl_tree2_row_expanded_is** (**Ewl_Tree2** *tree, void *data, unsigned int row)
Checks if row is expanded in data of tree.
- **Ewl_Tree_Selection_Type** **ewl_tree2_selection_type_get** (**Ewl_Tree2** *tree)
Get the selection type from the tree.
- void **ewl_tree2_selection_type_set** (**Ewl_Tree2** *tree, **Ewl_Tree_Selection_Type** type)
Set the mode of the tree.

6.75.3 Define Documentation

6.75.3.1 #define EWL_TREE2(t) ((Ewl_Tree2 *)t)

Typecasts a pointer to an **Ewl_Tree** (p. 769) pointer.

6.75.3.2 #define EWL_TREE2_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE2_TYPE))

Returns TRUE if the widget is an **Ewl_Tree2** (p. 771), FALSE otherwise

6.75.3.3 `#define EWL_TREE2_NODE(n) ((Ewl_Tree2_Node *)n)`

6.75.3.4 `#define EWL_TREE2_NODE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE2_NODE_TYPE))`

Returns TRUE if the widget is an **Ewl_Tree2_Node** (p. 774), FALSE otherwise

6.75.3.5 `#define EWL_TREE2_NODE_TYPE "node"`

6.75.3.6 `#define EWL_TREE2_TYPE "tree2"`

The type name for the Ewl_ widget

6.75.4 Typedef Documentation

6.75.4.1 `typedef struct Ewl_Tree2 Ewl_Tree2`

The **Ewl_Tree2** (p. 771) widget

6.75.4.2 `typedef struct Ewl_Tree2_Node Ewl_Tree2_Node`

Ewl_Tree2_Node (p. 774)

6.75.5 Function Documentation

6.75.5.1 void ewl_tree2_cb_column_sort (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.75.5.2 void ewl_tree2_cb_configure (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.75.5.3 void ewl_tree2_cb_destroy (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.75.5.4 void ewl_tree2_cb_node_child_add (Ewl_Container * *c*, Ewl_Widget * *w*)

6.75.5.5 void ewl_tree2_cb_node_child_del (Ewl_Container * *c*, Ewl_Widget * *w*, int *idx*)

6.75.5.6 void ewl_tree2_cb_node_child_hide (Ewl_Container * *c*, Ewl_Widget * *w*)

6.75.5.7 void ewl_tree2_cb_node_child_show (Ewl_Container * *c*, Ewl_Widget * *w*)

6.75.5.8 void ewl_tree2_cb_node_configure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.75.5.9 void ewl_tree2_cb_node_realize (Ewl_Widget * *w*, void * *ev*, void * *data*)

6.75.5.10 void ewl_tree2_cb_node_resize (Ewl_Container * *c*, Ewl_Widget * *w*, int *size*, Ewl_Orientation *o*)

6.75.5.11 void ewl_tree2_cb_node_toggle (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)

6.75.5.12 unsigned int ewl_tree2_column_count_get (Ewl_Tree2 * *tree*)

Retrives the number of columns in the tree.

Parameters:

tree,: The tree to work with

Returns:

Returns the number of columns in the tree

6.75.5.13 void ewl_tree2_column_count_set (Ewl_Tree2 * *tree*, unsigned int *count*)

Sets the number of columns in the tree.

Parameters:

tree,: The tree to work with

count,: The number of columns in the tree

Returns:

Returns no value

6.75.5.14 Ewl_View* ewl_tree2_content_view_get (Ewl_Tree2 * *tree*)

Retrives the view used to generate the tree content area.

Parameters:

tree,: The tree to work with

Returns:

Returns the view used to generate the content area

6.75.5.15 void ewl_tree2_content_view_set (Ewl_Tree2 * *tree*, Ewl_View * *view*)

Sets the view to use to generate the content area.

Parameters:

tree,: the tree to work with

view,: The view to set to generate the content area

Returns:

Returns no value

6.75.5.16 Ewl_Widget* ewl_tree2_content_widget_get (Ewl_Tree2 * *tree*)

Retrieves the widget containing the tree rows.

Parameters:

tree,: The tree to work with

Returns:

Returns the widget that contains the tree rows

6.75.5.17 unsigned int ewl_tree2_fixed_rows_get (Ewl_Tree2 * *tree*)

Retrieve the fixed row size of the tree.

Parameters:

tree,: The tree to get the fixed row flag from

Returns:

Returns the current fixed row flag of the tree

6.75.5.18 void ewl_tree2_fixed_rows_set (Ewl_Tree2 * *tree*, unsigned int *fixed*)

Set the fixed row size of the tree.

Parameters:

tree,: The tree to set the fixed row flag into
fixed,: The fixed row flag to set into the tree

Returns:

Returns no value.

6.75.5.19 unsigned int ewl_tree2_headers_visible_get (Ewl_Tree2 * *tree*)

Retrieve if the header is visible in the tree.

Parameters:

tree,: The tree to get the header visibility from

Returns:

Returns the current header visibility of the tree

6.75.5.20 void ewl_tree2_headers_visible_set (Ewl_Tree2 * *tree*, unsigned char *visible*)

Toggle if the header is visible in the tree.

Parameters:

tree,: The tree to toggle the header visibility
visible,: The visibility to set the tree to (TRUE == on, FALSE == off)

Returns:

Returns no value

6.75.5.21 int ewl_tree2_init (Ewl_Tree2 * *tree*)

Initialize the contents of a tree widget.

Parameters:

tree,: the tree widget to be initialized

Returns:

Returns TRUE on success, FALSE on failure.

The contents of the tree widget *tree* are initialized to their defaults.

6.75.5.22 Ewl_Widget* ewl_tree2_new (void)

Allocate and initialize a new tree widget.

Returns:

Returns NULL on failure, a new tree widget on success.

6.75.5.23 void ewl_tree2_node_collapse (Ewl_Tree2_Node * node)**6.75.5.24 void ewl_tree2_node_expand (Ewl_Tree2_Node * node)****6.75.5.25 unsigned int ewl_tree2_node_expandable_get (Ewl_Tree2_Node * node)****6.75.5.26 void ewl_tree2_node_expandable_set (Ewl_Tree2_Node * node, void * data)****6.75.5.27 unsigned int ewl_tree2_node_expanded_is (Ewl_Tree2_Node * node)****6.75.5.28 int ewl_tree2_node_init (Ewl_Tree2_Node * node)****6.75.5.29 Ewl_Widget* ewl_tree2_node_new (void)****6.75.5.30 void ewl_tree2_row_collapse (Ewl_Tree2 * tree, void * data, unsigned int row)**

Sets the given *row* to collapsed for the given *data* in *tree*.

Parameters:

tree,: The tree to collapse the row of

data,: The data that contains the collapsed row

row,: The row to collapse

Returns:

Returns no value

6.75.5.31 void ewl_tree2_row_expand (Ewl_Tree2 * tree, void * data, unsigned int row)

When the tree displays the data in *data* it will expand the give *row*. This *data* is the parent of the expansion row.

Parameters:

tree,: The tree to set the expansion into

data,: The data that contains the expansion

row,: The row to expand

Returns:

Returns no value

6.75.5.32 `unsigned int ewl_tree2_row_expanded_is (Ewl_Tree2 * tree, void * data, unsigned int row)`

Checks if *row* is expanded in *data* of *tree*.

Parameters:

tree,: The tree to work with

data,: The set of data to work with

row,: The row to check

Returns:

Returns TRUE if the given row is expanded, FALSE otherwise

6.75.5.33 `Ewl_Tree_Selection_Type ewl_tree2_selection_type_get (Ewl_Tree2 * tree)`

Get the selection type from the tree.

Parameters:

tree,: The tree to get the mode from

Returns:

Returns the current Ewl_Tree_Selection_Type of the tree

6.75.5.34 `void ewl_tree2_selection_type_set (Ewl_Tree2 * tree, Ewl_Tree_Selection_Type type)`

Set the mode of the tree.

Parameters:

tree,: The **Ewl_Tree** (p. 769) to set the mode into

type,: The Ewl_Tree_Selection_Mode to set into the tree

Returns:

Returns no value.

6.76 Ewl_Tree2_View: A tree view

6.76.1 Detailed Description

The tree view class.

Remarks:

Inherits from **Ewl_Box** (p. 586).

Data Structures

- struct **Ewl_Tree2_View**

*Inherits from **Ewl_Box** (p. 586) and provides the base information for a tree view.*

Defines

- `#define EWL_TREE2_VIEW(v) ((Ewl_Tree2_View *)v)`
- `#define EWL_TREE2_VIEW_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE2_VIEW_TYPE))`
- `#define EWL_TREE2_VIEW_TYPE "tree2_view"`

Typedefs

- `typedef Ewl_Tree2_View Ewl_Tree2_View`

Functions

- `int ewl_tree2_view_init (Ewl_Tree2_View *v)`
*Initializes an **Ewl_Tree2_View** (p. 776) to default values.*
- `Ewl_Tree2 * ewl_tree2_view_tree2_get (Ewl_Tree2_View *v)`
Retrieves the tree parent set into the view.
- `void ewl_tree2_view_tree2_set (Ewl_Tree2_View *v, Ewl_Tree2 *t)`
Sets the t tree into the v view.

6.76.2 Define Documentation

6.76.2.1 `#define EWL_TREE2_VIEW(v) ((Ewl_Tree2_View *)v)`

Typecasts a pointer to an **Ewl_Tree2_View** (p. 776) pointer

6.76.2.2 `#define EWL_TREE2_VIEW_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE2_VIEW_TYPE))`

Returns TRUE if the widget is an **Ewl_Tree2_View** (p. 776), FALSE otherwise

6.76.2.3 `#define EWL_TREE2_VIEW_TYPE "tree2_view"`

The type name for `Ewl_Tree2_View` (p. 776)

6.76.3 Typedef Documentation

6.76.3.1 `typedef struct Ewl_Tree2_View Ewl_Tree2_View`

A View on the tree

6.76.4 Function Documentation

6.76.4.1 `int ewl_tree2_view_init (Ewl_Tree2_View * v)`

Initializes an `Ewl_Tree2_View` (p. 776) to default values.

Parameters:

v,: The view to initialize

Returns:

Returns TRUE on success or FALSE on failure

6.76.4.2 `Ewl_Tree2* ewl_tree2_view_tree2_get (Ewl_Tree2_View * v)`

Retrieves the tree parent set into the view.

Parameters:

v,: The view to work with

Returns:

Returns the `Ewl_Tree2` (p. 771) parent

6.76.4.3 `void ewl_tree2_view_tree2_set (Ewl_Tree2_View * v, Ewl_Tree2 * t)`

Sets the *t* tree into the *v* view.

Parameters:

v,: The view to work with

t,: The tree to set

Returns:

Returns no value

6.77 Ewl_Tree2_View_Plain: The plain tree view

6.77.1 Detailed Description

Defines a plain view into the tree just listing rows.

Remarks:

Inherits from **Ewl_Tree2_View** (p. 776).

Data Structures

- struct **Ewl_Tree2_View_Plain**

*Inherits from **Ewl_Tree2_View** (p. 776) and provides a plain tree layout.*

Defines

- `#define EWL_TREE2_VIEW_PLAIN(tv) ((Ewl_Tree2_View_Plain*)tv)`
- `#define EWL_TREE2_VIEW_PLAIN_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE2_VIEW_PLAIN_TYPE))`
- `#define EWL_TREE2_VIEW_PLAIN_TYPE "tree2_view_plain"`

Typedefs

- `typedef Ewl_Tree2_View_Plain Ewl_Tree2_View_Plain`

Functions

- `Ewl_View * ewl_tree2_view_plain_get (void)`
*Retrieves a shared **Ewl_Tree2_View_Plain** (p. 778) widget.*
- `int ewl_tree2_view_plain_init (Ewl_Tree2_View_Plain *tv)`
*Initializes an **Ewl_Tree2_View_Plain** (p. 778) to default values.*
- `Ewl_Widget * ewl_tree2_view_plain_new (void)`
*Creates a new **Ewl_Tree2_View_Plain_View**.*

6.77.2 Define Documentation

6.77.2.1 `#define EWL_TREE2_VIEW_PLAIN(tv) ((Ewl_Tree2_View_Plain*)tv)`

Typecasts a pointer to an **Ewl_Tree2_View_Plain** (p. 778) pointer

6.77.2.2 `#define EWL_TREE2_VIEW_PLAIN_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE2_VIEW_PLAIN_TYPE))`

Returns TRUE if the widget is an **Ewl_Tree2_View_Plain** (p. 778), FALSE otherwise

6.77.2.3 `#define EWL_TREE2_VIEW_PLAIN_TYPE "tree2_view_plain"`

The `Ewl_Tree2_View_Plain` (p. 778) type

6.77.3 Typedef Documentation

6.77.3.1 `typedef struct Ewl_Tree2_View_Plain Ewl_Tree2_View_Plain`

`Ewl_Tree2_View_Plain` (p. 778)

6.77.4 Function Documentation

6.77.4.1 `Ewl_View* ewl_tree2_view_plain_get (void)`

Retrieves a shared `Ewl_Tree2_View_Plain` (p. 778) widget.

Returns:

Returns a default `Ewl_Tree2_View_Plain` (p. 778) struct

6.77.4.2 `int ewl_tree2_view_plain_init (Ewl_Tree2_View_Plain * tv)`

Initializes an `Ewl_Tree2_View_Plain` (p. 778) to default values.

Parameters:

tv,: The tree view to work with

Returns:

Returns TRUE on success or FALSE on failure

6.77.4.3 `Ewl_Widget* ewl_tree2_view_plain_new (void)`

Creates a new `Ewl_Tree2_View_Plain_View`.

Returns:

Returns a new `Ewl_Widget` (p. 785) on success or NULL on failure

6.78 Ewl_Tree2_View_Scrolled The scrolled tree view

6.78.1 Detailed Description

A scrolled view for the tree.

Remarks:

Inherits from **Ewl_Box** (p. 586).

Data Structures

- struct **Ewl_Tree2_View_Scrolled**

*Inherits from **Ewl_Box** (p. 586) and provides a scrolled tree layout.*

Defines

- `#define EWL_TREE2_VIEW_SCROLLED(tv) ((Ewl_Tree2_View_Scrolled*)tv)`
- `#define EWL_TREE2_VIEW_SCROLLED_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE2_VIEW_SCROLLED_TYPE))`
- `#define EWL_TREE2_VIEW_SCROLLED_TYPE "tree2_view_scrolled"`

Typedefs

- `typedef Ewl_Tree2_View_Scrolled Ewl_Tree2_View_Scrolled`

Functions

- `Ewl_View * ewl_tree2_view_scrolled_get (void)`
Retrieves the view for this widget.
- `int ewl_tree2_view_scrolled_init (Ewl_Tree2_View_Scrolled *tv)`
Initializes a tree view to default values.
- `Ewl_Widget * ewl_tree2_view_scrolled_new (void)`
*Creates and initializes a new **Ewl_Tree2_View_Scrolled** (p. 779) widget.*
- `unsigned int ewl_tree2_view_scrolled_scroll_headers_get (Ewl_Tree2_View *view)`
Retrieves if the tree headers will be scrolled or not.
- `void ewl_tree2_view_scrolled_scroll_headers_set (Ewl_Tree2_View *view, unsigned int scroll)`
Sets if the headers should be scrolled with the tree.

6.78.2 Define Documentation

6.78.2.1 `#define EWL_TREE2_VIEW_SCROLLED(tv) ((Ewl_Tree2_View_Scrolled*)tv)`

Typecasts a pointer to an `Ewl_Tree2_View_Scrolled` (p. 779) pointer

6.78.2.2 `#define EWL_TREE2_VIEW_SCROLLED_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE2_VIEW_SCROLLED_TYPE))`

Returns TRUE if the widget is an `Ewl_Tree2_View_Scrolled` (p. 779), FALSE otherwise

6.78.2.3 `#define EWL_TREE2_VIEW_SCROLLED_TYPE "tree2_view_scrolled"`

The type name

6.78.3 Typedef Documentation

6.78.3.1 `typedef struct Ewl_Tree2_View_Scrolled Ewl_Tree2_View_Scrolled`

`Ewl_Tree2_View_Scrolled` (p. 779)

6.78.4 Function Documentation

6.78.4.1 `Ewl_View* ewl_tree2_view_scrolled_get (void)`

Retrieves the view for this widget.

Returns:

Returns the view for this widget

6.78.4.2 `int ewl_tree2_view_scrolled_init (Ewl_Tree2_View_Scrolled * tv)`

Initialies a tree view to default values.

Parameters:

tv,: The view to work with

Returns:

Returns TRUE on success or FALSE on failure

6.78.4.3 `Ewl_Widget* ewl_tree2_view_scrolled_new (void)`

Creates and initializes a new `Ewl_Tree2_View_Scrolled` (p. 779) widget.

Returns:

Returns a new **Ewl_Widget** (p. 785) on success or NULL on failure

6.78.4.4 unsigned int ewl_tree2_view_scrolled_scroll_headers_get
(Ewl_Tree2_View * *view*)

Retrieves if the tree headers will be scrolled or not.

Parameters:

view,: The view to work with

Returns:

Returns no value

6.78.4.5 void ewl_tree2_view_scrolled_scroll_headers_set (Ewl_Tree2_View *
view, unsigned int *scroll*)

Sets if the headers should be scrolled with the tree.

Parameters:

view,: The view to work with

scroll,: If the headers should be scrolled

Returns:

Returns no value

6.79 Ewl_View: A data view

6.79.1 Detailed Description

Defines the callbacks for setting up the widgets based on the data returned from the Model. Create widgets, set data on widgets, calculate sizing, minimize number of widgets.

Data Structures

- struct **Ewl_View**
The view function pointers.

Defines

- #define **EWL_VIEW**(view) ((**Ewl_View** *)view)
- #define **EWL_VIEW_EXPANSION_VIEW_FETCH_GET**(f) ((**Ewl_View_Expansion_View_Fetch**)f)
- #define **EWL_VIEW_HEADER_GET**(f) ((**Ewl_View_Header_Fetch**)f)
- #define **EWL_VIEW_WIDGET_FETCH**(f) ((**Ewl_View_Widget_Fetch**)f)

Typedefs

- typedef **Ewl_View** **Ewl_View**
- typedef **Ewl_View** (*) **Ewl_View_Expansion_View_Fetch** (void *data, int row)
- typedef **Ewl_Widget** (*) **Ewl_View_Header_Fetch** (void *data, int column)
- typedef **Ewl_Widget** (*) **Ewl_View_Widget_Fetch** (void *data, int row, int col)

Functions

- **Ewl_View** * **ewl_view_clone** (**Ewl_View** *src)
*Creates a new **Ewl_View** (p. 784) object.*
- **Ewl_View_Expansion_View_Fetch** **ewl_view_expansion_view_fetch_get** (**Ewl_View** *v)
Gets the expansion view fetch callback from the view.
- void **ewl_view_expansion_view_fetch_set** (**Ewl_View** *v, **Ewl_View_Expansion_View_Fetch** f)
Sets the expansion view fetch callback into the view.
- **Ewl_View_Header_Fetch** **ewl_view_header_fetch_get** (**Ewl_View** *v)
Gets the header fetch callback from the view.
- void **ewl_view_header_fetch_set** (**Ewl_View** *v, **Ewl_View_Header_Fetch** f)
Sets the header fetch callback into the view.
- int **ewl_view_init** (**Ewl_View** *view)
*Initializes an **Ewl_View** (p. 784) object to default values.*

- **Ewl_View * ewl_view_new** (void)
*Creates a new **Ewl_View** (p. 784) object.*
- **Ewl_View_Widget_Fetch ewl_view_widget_fetch_get** (Ewl_View *view)
Get the constructor set on this view.
- **void ewl_view_widget_fetch_set** (Ewl_View *view, Ewl_View_Widget_Fetch construct)
This will set the given widget fetch callback into the view.

6.79.2 Define Documentation

6.79.2.1 #define EWL_VIEW(view) ((Ewl_View *)view)

Typecasts a pointer to an **Ewl_View** (p. 784) pointer.

6.79.2.2 #define EWL_VIEW_EXPANSION_VIEW_FETCH_GET(f) ((Ewl_View_Expansion_View_Fetch)f)

View callback to get a view for expanded data on a row

6.79.2.3 #define EWL_VIEW_HEADER_GET(f) ((Ewl_View_Header_Fetch)f)

View callback to get the header for a given column

6.79.2.4 #define EWL_VIEW_WIDGET_FETCH(f) ((Ewl_View_Widget_Fetch)f)

View callback for the constructor for a data row

6.79.3 Typedef Documentation

6.79.3.1 typedef struct Ewl_View Ewl_View

The **Ewl_View** (p. 784) type

6.79.3.2 typedef Ewl_View*(*) Ewl_View_Expansion_View_Fetch(void *data, int row)

A typedef to shorten the definition of the view_expansion_view_fetch callbacks.

6.79.3.3 typedef Ewl_Widget*(*) Ewl_View_Header_Fetch(void *data, int column)

A typedef to shorten the definition of the view_header_fetch callbacks.

6.79.3.4 `typedef Ewl_Widget*(*) Ewl_View_Widget_Fetch(void *data, int row, int col)`

A typedef to shorten the definition of the `view_widget_fetch` callbacks.

6.79.4 Function Documentation

6.79.4.1 `Ewl_View* ewl_view_clone (Ewl_View * src)`

Creates a new **Ewl_View** (p. 784) object.

Parameters:

src,: An existing view to copy as a basis for a new view

Returns:

Returns a new **Ewl_View** (p. 784) object on success or NULL on failure

6.79.4.2 `Ewl_View_Expansion_View_Fetch ewl_view_expansion_view_fetch_get (Ewl_View * v)`

Gets the expansion view fetch callback from the view.

Parameters:

v,: The **Ewl_View** (p. 784) to get the `Ewl_View_Expansion_View_Fetch` function from

Returns:

Returns the `Ewl_View_Expansion_View_Fetch` callback set on the view, or NULL on failure.

6.79.4.3 `void ewl_view_expansion_view_fetch_set (Ewl_View * v, Ewl_View_Expansion_View_Fetch f)`

Sets the expansion view fetch callback into the view.

Parameters:

v,: The **Ewl_View** (p. 784) to set the `expansion_view_fetch` callback on

f,: The `Ewl_View_Expansion_View_Fetch` callback

Returns:

Returns no value.

6.79.4.4 Ewl_View_Header_Fetch `ewl_view_header_fetch_get (Ewl_View * v)`

Gets the header fetch callback from the view.

Parameters:

v,: The **Ewl_View** (p. 784) to get the `Ewl_View_Header_Fetch` function from

Returns:

Returns the `Ewl_View_Header_Fetch` callback set on the view, or NULL on failure.

6.79.4.5 void ewl_view_header_fetch_set `(Ewl_View * v,
Ewl_View_Header_Fetch f)`

Sets the header fetch callback into the view.

Parameters:

v,: The **Ewl_View** (p. 784) to set the `header_fetch` callback on

f,: The `Ewl_View_Header_Fetch` callback

Returns:

Returns no value.

6.79.4.6 int ewl_view_init `(Ewl_View * view)`

Initializes an **Ewl_View** (p. 784) object to default values.

Parameters:

view,: The **Ewl_View** (p. 784) to initialize

Returns:

Returns TRUE on success or FALSE on failure

6.79.4.7 Ewl_View* ewl_view_new `(void)`

Creates a new **Ewl_View** (p. 784) object.

Returns:

Returns a new **Ewl_View** (p. 784) object on success or NULL on failure

6.79.4.8 Ewl_View_Widget_Fetch `ewl_view_widget_fetch_get (Ewl_View * v)`

Get the constructor set on this view.

Parameters:

v,: The **Ewl_View** (p. 784) to get the constructor from

Returns:

Returns the `Ewl_View_Constructor` set into the view or `NULL` if none set.

6.79.4.9 void ewl_view_widget_fetch_set (`Ewl_View * v`,
`Ewl_View_Widget_Fetch fetch`)

This will set the given widget fetch callback into the view.

Parameters:

v,: The **Ewl_View** (p. 784) to set the widget fetch callback into

fetch,: The `Ewl_View_Widget_Fetch` to set into the view

Returns:

Returns no value.

6.80 Ewl_Widget: The Parent Widget Class Common to All Widgets

6.80.1 Detailed Description

Defines the **Ewl_Widget** (p. 785) class and its accessor/modifier functions.

The **Ewl_Widget** (p. 785) extends the **Ewl_Object** (p. 709) to provide the basic facilities necessary for widgets to interact with the end user. This includes basic callbacks for input events, window information changes, and drawing to the display.

Remarks:

Inherits from **Ewl_Object** (p. 709).

6.80.2 Tutorial

Small as small can be (originally at <http://everburning.com/news/small-as-small-can-be>)

Whats the minimum amount of work you need to do to create your own EWL widget? Just want something you can build on but dont know where to start?

Well, hopefully this should give you the base for starting your widget. Assuming you're creating a widget called My_Widget, the EWL convention is to have a my_widget.c and my_widget.h files. There are only a couple things you need to implement to get a working widget.

First, my_widget.h.

```
#ifndef MY_WIDGET_H
#define MY_WIDGET_H

#include <Ewl.h>

#define MY_WIDGET(w) ((My_Widget *)w)
#define MY_WIDGET_TYPE "my_widget"

typedef struct My_Widget My_Widget;
struct My_Widget
{
    Ewl_Widget widget;
};

Ewl_Widget *my_widget_new(void);
int my_widget_init(My_Widget *w);

#endif
```

That wasn't so bad. What have we got? Well, the MY_WIDGET(w) define gives us a simple macro to cast other widgets to our widget. The second define, MY_WIDGET_TYPE, is a simple macro containing the type name of the widget. Well use that a bit later (and in any type checking we add to our widget.)

We then create the widget structure. In this case were inheriting from **Ewl_Widget** (p. 785) so its the first item in our struct (and not a pointer, thats important). This is how EWLs inheritance works. The widget you're inheriting from is the first item in the struct and not a pointer. You will now be able to call any of the methods of the inherited class on the new class.

We then declare two methods. The convention in EWL is that the _new() function always takes no parameters (void). There is also always a _init() function that takes the widget as its only parameter and returns an int, if the initialization succeeded or failed.

With that out of the way, lets take a look at `my_widget.c`.

```
#include "my_widget.h"

Ewl_Widget *
my_widget_new(void)
{
    Ewl_Widget *w;

    w = calloc(1, sizeof(My_Widget));
    if (!w) return NULL;

    if (!my_widget_init(MY_WIDGET(w)))
    {
        free(w);
        return NULL;
    }
    return w;
}

int
my_widget_init(My_Widget *w)
{
    if (!ewl_widget_init(EWL_WIDGET(w)))
        return 0;

    ewl_widget_appearance_set(EWL_WIDGET(w), MY_WIDGET_TYPE);
    ewl_widget_inherit(EWL_WIDGET(w), MY_WIDGET_TYPE);

    return 1;
}
```

Thats pretty simple. We create a new widget, initialize it and thats about it. In `my_widget_init()` we make sure we call **`ewl_widget_init()`** (p.555) as thats the widget we are inheriting from and we then set our inheritance and appearance strings (notice the use of our type define from earlier).

With that you've got a simple widget. It doesn't do much, but it exists. Build on as you will.

Data Structures

- struct **Ewl_Attach_List**

A list of things attached to a widget.

- struct **Ewl_Callback_Chain**

The callback chain contains the length, mask and information on the list.

- struct **Ewl_Color_Set**

Contains an RGBA set of colours.

- struct **Ewl_Pair**

Contains a key and a value pair.

- struct **Ewl_Pair_List**

Contains a list of key value pairs.

- struct **Ewl_Widget**

*Inherits from **Ewl_Object** (p. 709) and extends to provide appearance, parent, and callback capabilities.*

Defines

- `#define EWL_PAIR(p) ((Ewl_Pair *)p)`
- `#define EWL_WIDGET(widget) ((Ewl_Widget *) widget)`
*Typecast a pointer to an **Ewl_Widget** (p. 785) pointer.*
- `#define EWL_WIDGET_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_WIDGET_TYPE))`
- `#define EWL_WIDGET_TYPE "widget"`

Typedefs

- `typedef Ewl_Attach_List Ewl_Attach_List`
- `typedef Ewl_Callback_Chain Ewl_Callback_Chain`
- `typedef Ewl_Color_Set Ewl_Color_Set`
- `typedef Ewl_Pair Ewl_Pair`
- `typedef Ewl_Pair_List Ewl_Pair_List`
- `typedef Ewl_Widget Ewl_Widget`
- `typedef void (*)(*) Ewl_Widget_Drag (void)`

Functions

- `char * ewl_widget_appearance_get (Ewl_Widget *w)`
Retrieve the appearance key of the widget.
- `const char * ewl_widget_appearance_part_text_get (Ewl_Widget *w, const char *part)`
Retrieve a copy of a parts current text.
- `void ewl_widget_appearance_part_text_set (Ewl_Widget *w, const char *part, const char *text)`
Change the text of the given theme part of a widget.
- `int ewl_widget_appearance_path_copy (Ewl_Widget *w, char *buf, int size)`
- `char * ewl_widget_appearance_path_get (Ewl_Widget *w)`
Retrieve the appearance path key of the widget.
- `int ewl_widget_appearance_path_size_get (Ewl_Widget *w)`
- `void ewl_widget_appearance_set (Ewl_Widget *w, const char *appearance)`
Change the appearance of the specified widget.
- `const char * ewl_widget_appearance_text_get (Ewl_Widget *w)`
Retrieve the text of the given theme part of a widget.
- `void ewl_widget_appearance_text_set (Ewl_Widget *w, const char *text)`

Change the text of the given theme part of a widget.

- void **ewl_widget_cb_configure** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_disable** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_enable** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_focus_in** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_focus_out** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_hide** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_mouse_down** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_mouse_in** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_mouse_move** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_mouse_out** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_mouse_up** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_obscure** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_realize** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_reparent** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_reveal** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_show** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_widget_cb_unrealize** (Ewl_Widget *w, void *ev_data, void *user_data)
- unsigned int **ewl_widget_clipped_is** (Ewl_Widget *w)

Checks if a widget clips it's theme object.

- void **ewl_widget_clipped_set** (Ewl_Widget *w, unsigned int val)

Marks whether the widget should be clipped at it's boundaries.

- void **ewl_widget_color_get** (Ewl_Widget *w, int *r, int *g, int *b, int *a)

Gets the colour settings of the widget.

- void **ewl_widget_color_set** (Ewl_Widget *w, int r, int g, int b, int a)

sets the colour of the widget

- void **ewl_widget_configure** (Ewl_Widget *widget)

Initiate configuring of the specified widget.

- void * **ewl_widget_data_del** (Ewl_Widget *w, void *k)

Remove the specified key / value pair from the widget and return the value.

- void * **ewl_widget_data_get** (Ewl_Widget *w, void *k)

retrieve the specified key / value pair from the widget

- void **ewl_widget_data_set** (Ewl_Widget *w, void *k, void *v)

Attach the specified key / value pair to the widget.

- void **ewl_widget_destroy** (Ewl_Widget *widget)

Destroy the specified widget.

- void **ewl_widget_disable** (Ewl_Widget *w)

Prevent a widget from receiving any events.

- void **ewl_widget_dnd_reset** (void)
Cancel any active dnd_wait state widgets.
- **Ewl_Widget * ewl_widget_drag_candidate_get** (void)
Accessor function for the current drag candidate widget.
- void **ewl_widget_draggable_set** (Ewl_Widget *w, unsigned int val, Ewl_Widget_Drag cb)
Set the draggable state, and setup any callbacks.
- void **ewl_widget_enable** (Ewl_Widget *w)
Re-enable a disabled widget.
- void **ewl_widget_focus_send** (Ewl_Widget *w)
Changes the keyboard focus to the widget w.
- unsigned int **ewl_widget_focusable_get** (Ewl_Widget *w)
Checks the focusable state of the widget.
- void **ewl_widget_focusable_set** (Ewl_Widget *w, unsigned int val)
Set if the given widget is focusable or not.
- **Ewl_Widget * ewl_widget_focused_get** (void)
Retrieve the currently focused widget.
- void **ewl_widget_free** (Ewl_Widget *w)
- void **ewl_widget_hide** (Ewl_Widget *widget)
Mark a widget as invisible.
- unsigned int **ewl_widget_ignore_focus_change_get** (Ewl_Widget *w)
Get if the widget is ignoring focus changes.
- void **ewl_widget_ignore_focus_change_set** (Ewl_Widget *w, unsigned int val)
Set if the widget should ignore focus changes.
- void **ewl_widget_inherit** (Ewl_Widget *widget, const char *type)
Appends the given inheritance to this widgets inheritance string.
- int **ewl_widget_init** (Ewl_Widget *w)
Initialize a widget to default values and callbacks.
- unsigned int **ewl_widget_internal_is** (Ewl_Widget *w)
- void **ewl_widget_internal_set** (Ewl_Widget *w, unsigned int val)
- int **ewl_widget_layer_priority_get** (Ewl_Widget *w)
Retrieve a widgets layer relative to it's parent.
- void **ewl_widget_layer_priority_set** (Ewl_Widget *w, int layer)
Set the relative layer to it's parent.

- **int ewl_widget_layer_top_get** (Ewl_Widget *w)
Returns if the widget will be drawn above all the others.
- **void ewl_widget_layer_top_set** (Ewl_Widget *w, int top)
set the widget to be layered above all other widgets
- **Ewl_Widget * ewl_widget_name_find** (const char *name)
Find a widget identified by a name.
- **const char * ewl_widget_name_get** (Ewl_Widget *w)
Get the name for the specified widget.
- **void ewl_widget_name_set** (Ewl_Widget *w, const char *name)
Name the specified widget.
- **void ewl_widget_obscure** (Ewl_Widget *w)
Indicate a widget is obscured.
- **unsigned int ewl_widget_onscreen_is** (Ewl_Widget *widget)
Checks if the given widget is currently on screen.
- **Ewl_Widget * ewl_widget_parent_get** (Ewl_Widget *w)
Retrieves the parent of the given widget.
- **int ewl_widget_parent_of** (Ewl_Widget *c, Ewl_Widget *w)
Determine if a widget is a parent of another widget.
- **void ewl_widget_parent_set** (Ewl_Widget *w, Ewl_Widget *p)
change the parent of the specified widget
- **void ewl_widget_print** (Ewl_Widget *w)
Prints info for debugging a widget's state information.
- **void ewl_widget_print_verbose** (Ewl_Widget *w)
Prints verbose info for debugging a widget's state information.
- **void ewl_widget_realize** (Ewl_Widget *widget)
Realize the specified widget.
- **void ewl_widget_reparent** (Ewl_Widget *widget)
initiate reparent of the specified widget
- **void ewl_widget_reveal** (Ewl_Widget *w)
Indicate a widget is revealed.
- **void ewl_widget_show** (Ewl_Widget *widget)
mark a widget as visible
- **void ewl_widget_state_set** (Ewl_Widget *w, const char *state, Ewl_State_Type flag)

Update the appearance of the widget to a state.

- `void ewl_widget_tab_order_append (Ewl_Widget *w)`
Changes the order in the embed so w receives focus first on tab.
- `void ewl_widget_tab_order_insert (Ewl_Widget *w, unsigned int idx)`
Changes the order in the embed so w receives focus first on tab.
- `void ewl_widget_tab_order_insert_after (Ewl_Widget *w, Ewl_Widget *after)`
Insert the given widget into the tab order after the after widget.
- `void ewl_widget_tab_order_insert_before (Ewl_Widget *w, Ewl_Widget *before)`
Inserts the widget into the tab order before the before widget.
- `void ewl_widget_tab_order_prepend (Ewl_Widget *w)`
Changes the order in the embed so w receives focus first on tab.
- `void ewl_widget_tab_order_remove (Ewl_Widget *w)`
Remove the widget from the tab order.
- `void ewl_widget_tree_print (Ewl_Widget *w)`
Prints to stdout the tree of widgets that are parents of a widget.
- `unsigned int ewl_widget_type_is (Ewl_Widget *widget, const char *type)`
Determine if the widget w has inherited from the type t.
- `void ewl_widget_unrealize (Ewl_Widget *w)`
Unrealize the specified widget.

6.80.3 Define Documentation

6.80.3.1 `#define EWL_PAIR(p) ((Ewl_Pair *)p)`

Typedcasts a pointer to an `Ewl_Pair` (p. 713) pointer

6.80.3.2 `#define EWL_WIDGET(widget) ((Ewl_Widget *) widget)`

Typecast a pointer to an `Ewl_Widget` (p. 785) pointer.

6.80.3.3 `#define EWL_WIDGET_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_WIDGET_TYPE))`

Returns TRUE if the widget is an `Ewl_Widget` (p. 785), FALSE otherwise

6.80.3.4 `#define EWL_WIDGET_TYPE "widget"`

The type name for the `Ewl_Widget` (p. 785) widget

6.80.4 Typedef Documentation

6.80.4.1 `typedef struct Ewl_Attach_List Ewl_Attach_List`

The attachment list

6.80.4.2 `typedef struct Ewl_Callback_Chain Ewl_Callback_Chain`

Callback chain contains a list and bitmask of chain properties.

6.80.4.3 `typedef struct Ewl_Color_Set Ewl_Color_Set`

A set of colours

6.80.4.4 `typedef struct Ewl_Pair Ewl_Pair`

A key/value pair set

6.80.4.5 `typedef struct Ewl_Pair_List Ewl_Pair_List`

A list of key value pairs

6.80.4.6 `typedef struct Ewl_Widget Ewl_Widget`

The class that all widgets should inherit. Provides reference to a parent widget/container, callbacks, and appearance information.

6.80.4.7 `typedef void*(*) Ewl_Widget_Drag(void)`

Function pointer for the Ewl widget drag

6.80.5 Function Documentation

6.80.5.1 `char* ewl_widget_appearance_get (Ewl_Widget * w)`

Retrieve the appearance key of the widget.

Parameters:

w,: the widget to retrieve the appearance key

Returns:

Returns a pointer to the appearance key string on success, NULL on failure.

6.80.5.2 `const char* ewl_widget_appearance_part_text_get (Ewl_Widget * w,
const char * part)`

Retrieve a copy of a parts current text.

Parameters:

w,: the widget whose text to retrieve

part,: the theme part name whose text to retrieve

Returns:

Returns NULL on failure, a copy of the current text on success.

Get the text of a given Edje-define TEXT part. This is for widgets whose Edje appearance defines TEXT parts, and enables each of those text parts to be retrieved independently.

The returned string will only be valid until the next time text is set on this part.

6.80.5.3 `void ewl_widget_appearance_part_text_set (Ewl_Widget * w, const
char * part, const char * text)`

Change the text of the given theme part of a widget.

Parameters:

w,: the widget whose text to change

part,: the theme part name whose text to change

text,: the new text to change to

Returns:

Returns no value.

Changes the text of a given Edje-define TEXT part. This is for widgets whose Edje appearance defines TEXT parts, and enables each of those text parts to be changed independently. The text value is recorded in a hash and reapplied if the theme is reloaded for this widget.

6.80.5.4 `int ewl_widget_appearance_path_copy (Ewl_Widget * w, char * buf,
int size)`**6.80.5.5** `char* ewl_widget_appearance_path_get (Ewl_Widget * w)`

Retrieve the appearance path key of the widget.

Parameters:

w,: the widget to retrieve the full appearance key

Returns:

Returns a pointer to the full appearance path string on success, NULL on failure.

6.80.5.6 `int ewl_widget_appearance_path_size_get (Ewl_Widget * w)`

6.80.5.7 `void ewl_widget_appearance_set (Ewl_Widget * w, const char *
appearance)`

Change the appearance of the specified widget.

Parameters:

w,: the widget to change the appearance

appearance,: the new key for the widgets appearance

Returns:

Returns no value.

Changes the key associated with the widgets appearance and calls the theme update callback to initiate the change.

6.80.5.8 `const char* ewl_widget_appearance_text_get (Ewl_Widget * w)`

Retrieve the text of the given theme part of a widget.

Parameters:

w,: the widget whose text to retrieve

Returns:

Returns the current text on success, NULL on failure.

Note, the returned value will only be valid until the next time `ewl_widget_appearance_text_set()` (p. 548) is called on this widget.

6.80.5.9 `void ewl_widget_appearance_text_set (Ewl_Widget * w, const char *
text)`

Change the text of the given theme part of a widget.

Parameters:

w,: the widget whose text to change

text,: the new text to change to

Returns:

Returns no value.

Changes the text of an Edge-define TEXT part. This is for widgets whose Edge appearance defines a TEXT part, and identifies it with with a data item called "/WIDGET/textpart". The text value is recorded in a hash and reapplied if the theme is reloaded for this widget.

- 6.80.5.10 void ewl_widget_cb_configure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.11 void ewl_widget_cb_disable (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.12 void ewl_widget_cb_enable (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.13 void ewl_widget_cb_focus_in (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.14 void ewl_widget_cb_focus_out (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.15 void ewl_widget_cb_hide (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.16 void ewl_widget_cb_mouse_down (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.17 void ewl_widget_cb_mouse_in (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.18 void ewl_widget_cb_mouse_move (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.19 void ewl_widget_cb_mouse_out (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.20 void ewl_widget_cb_mouse_up (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.21 void ewl_widget_cb_obscure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.22 void ewl_widget_cb_realize (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.23 void ewl_widget_cb_reparent (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.24 void ewl_widget_cb_reveal (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.25 void ewl_widget_cb_show (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.26 void ewl_widget_cb_unrealize (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.80.5.27 unsigned int ewl_widget_clipped_is (Ewl_Widget * *w*)

Checks if a widget clips it's theme object.

Parameters:

w,: the widget to check if it clips it's theme object

Returns:

Returns TRUE if the widget clips, otherwise FALSE.

6.80.5.28 void ewl_widget_clipped_set (Ewl_Widget * *w*, unsigned int *val*)

Marks whether the widget should be clipped at it's boundaries.

Parameters:

w,: the widget to mark as unclipped

val,: the state of the clipping flag

Returns:

Returns no value.

6.80.5.29 void ewl_widget_color_get (Ewl_Widget * *w*, int * *r*, int * *g*, int * *b*, int * *a*)

Gets the colour settings of the widget.

Parameters:

w,: The widget to get the colour from

r,: Where to put the red value

g,: Where to put the green value

b,: Where to put the blue value

a,: Where to put the alpha value

Returns:

Returns no value

6.80.5.30 void ewl_widget_color_set (Ewl_Widget * *w*, int *r*, int *g*, int *b*, int *a*)

sets the colour of the widget

Parameters:

w,: The widget to set the color of

r,: The red value

g,: The green value

b,: The blue value

a,: The alpha value

Returns:

Returns no value

6.80.5.31 void ewl_widget_configure (Ewl_Widget * *w*)

Initiate configuring of the specified widget.

Parameters:

w,: the widget to configure

Returns:

Returns no value.

The configure callback is triggered for the specified widget, this should adjust the widgets size and position.

6.80.5.32 void* ewl_widget_data_del (Ewl_Widget * *w*, void * *k*)

Remove the specified key / value pair from the widget and return the value.

Parameters:

w,: the widget that owns the key value pair

k,: the key that is associated with the data

Returns:

Returns the deleted value.

Removes a key / value pair with *k* as the key from the specified widget *w* and return the value. NULL is returned if there is no stored data or if an error occurs.

6.80.5.33 void* ewl_widget_data_get (Ewl_Widget * *w*, void * *k*)

retrieve the specified key / value pair from the widget

Parameters:

w,: the widget that owns the key value pair

k,: the key that is associated with the data

Returns:

Returns the value associated with *k* on success, NULL on failure.

Retrieves a key / value pair with *k* as the key from the specified widget *w*.

6.80.5.34 void ewl_widget_data_set (Ewl_Widget * *w*, void * *k*, void * *v*)

Attach the specified key / value pair to the widget.

Parameters:

w,: the widget to own the key value pair

k,: the key that is associated with the data

v,: the data that is to be tracked

Returns:

Returns no value.

Assigns a key / value pair with *k* as the key and *v* as the value to the specified widget *w*.

6.80.5.35 void ewl_widget_destroy (Ewl_Widget * *w*)

Destroy the specified widget.

Parameters:

w,: the widget to be destroyed

Returns:

Returns no value.

The widget calls it's destroy callback to do any clean up necessary and then free's the widget.

6.80.5.36 void ewl_widget_disable (Ewl_Widget * *w*)

Prevent a widget from receiving any events.

Parameters:

w,: the widget to disable

Returns:

Returns no value. Disables a specified widget.

This prevents that widget from receiving any user input events.

6.80.5.37 void ewl_widget_dnd_reset (void)

Cancel any active dnd_wait state widgets.

Returns:

none

6.80.5.38 Ewl_Widget* ewl_widget_drag_candidate_get (void)

Accessor function for the current drag candidate widget.

Returns:

Returns the current widget that has been clicked or moved (i.e. drag_wait)

6.80.5.39 void ewl_widget_draggable_set (Ewl_Widget * *w*, unsigned int *val*, Ewl_Widget_Drag *cb*)

Set the draggable state, and setup any callbacks.

Parameters:

w,: the widget to set draggable state
val,: the true/false state of draggable
cb,: The drag callback to set

Returns:

Returns no value.

6.80.5.40 void ewl_widget_enable (Ewl_Widget * *w*)

Re-enable a disabled widget.

Parameters:

w,: the widget to re-enable

Returns:

Returns no value.

Re-enables a previously disabled widget.

6.80.5.41 void ewl_widget_focus_send (Ewl_Widget * *w*)

Changes the keyboard focus to the widget *w*.

Parameters:

w,: the widget to receive keyboard focus

Returns:

Returns no value.

6.80.5.42 unsigned int ewl_widget_focusable_get (Ewl_Widget * *w*)

Checks the focusable state of the widget.

Parameters:

w,: The widget to get the focusable state from

Returns:

Returns TRUE if the widget is focusable, FALSE otherwise

6.80.5.43 void ewl_widget_focusable_set (Ewl_Widget * *w*, unsigned int *val*)

Set if the given widget is focusable or not.

Parameters:

w,: The widget to set the focusable values

val,: The focusable value to set

Returns:

Returns no value

6.80.5.44 Ewl_Widget* ewl_widget_focused_get (void)

Retrieve the currently focused widget.

Returns:

Returns the currently focused widget.

6.80.5.45 void ewl_widget_free (Ewl_Widget * *w*)**6.80.5.46 void ewl_widget_hide (Ewl_Widget * *w*)**

Mark a widget as invisible.

Parameters:

w,: the widget to be marked as invisible

Returns:

Returns no value.

Marks the widget as invisible so that it will not be displayed the next time through the rendering loop.

6.80.5.47 unsigned int ewl_widget_ignore_focus_change_get (Ewl_Widget * *w*)

Get if the widget is ignoring focus changes.

Parameters:

w,: The widget to check if it blocks tab focus

Returns:

Returns TRUE if the widget blocks tab focus, FALSE otherwise.

6.80.5.48 void ewl_widget_ignore_focus_change_set (Ewl_Widget * *w*, unsigned int *val*)

Set if the widget should ignore focus changes.

Parameters:

w,: The widget to set if it accepts or blocks focus changes
val,: TRUE or FALSE on if this widget blocks tabbing off

Returns:

Returns no value.

6.80.5.49 void ewl_widget_inherit (Ewl_Widget * *widget*, const char * *inherit*)

Appends the given inheritance to this widgets inheritance string.

Parameters:

widget,: the widget to set the inheritance on
inherit,: the string to append to the inheritance

Returns:

Returns no value.

6.80.5.50 int ewl_widget_init (Ewl_Widget * *w*)

Initialize a widget to default values and callbacks.

Parameters:

w,: the widget to initialize

Returns:

Returns TRUE on success, FALSE on failure.

The widget *w* is initialized to default values and is assigned the default callbacks.

6.80.5.51 unsigned int ewl_widget_internal_is (Ewl_Widget * *w*)

6.80.5.52 void ewl_widget_internal_set (Ewl_Widget * *w*, unsigned int *val*)

6.80.5.53 int ewl_widget_layer_priority_get (Ewl_Widget * *w*)

Retrieve a widgets layer relative to it's parent.

Parameters:

w,: the widget to retrieve current relative layer

Returns:

Returns a widgets current layer relative to it's parent.

6.80.5.54 void ewl_widget_layer_priority_set (Ewl_Widget * *w*, int *layer*)

Set the relative layer to it's parent.

Parameters:

w,: the widget to change relative layers

layer,: the new relative layer of the widget

Returns:

Returns no value.

Changes the current layer of *w* relative to it's parent. The default value is 0.

6.80.5.55 int ewl_widget_layer_top_get (Ewl_Widget * *w*)

Returns if the widget will be drawn above all the others.

Parameters:

w,: the widget to get the top value

Returns:

Returns TRUE or FALSE

6.80.5.56 void ewl_widget_layer_top_set (Ewl_Widget * *w*, int *top*)

set the widget to be layered above all other widgets

Parameters:

w,: the widget to set the top value

top,: TRUE or FALSE

Returns:

Returns no value.

This set the widget to be layered above all other widgets.

6.80.5.57 Ewl_Widget* ewl_widget_name_find (const char * *name*)

Find a widget identified by a name.

Parameters:

name,: the name of the widget to retrieve

Returns:

Returns an pointer a matched widget on success.

6.80.5.58 `const char* ewl_widget_name_get (Ewl_Widget * w)`

Get the name for the specified widget.

Parameters:

w,: the widget to retrieve the name

Returns:

Returns an pointer to an allocated name string on success.

6.80.5.59 `void ewl_widget_name_set (Ewl_Widget * w, const char * name)`

Name the specified widget.

Parameters:

w,: the widget to name

name,: the new name for the widget

Returns:

Returns no value.

6.80.5.60 `void ewl_widget_obscure (Ewl_Widget * w)`

Indicate a widget is obscured.

Parameters:

w,: the widget to mark as obscured

Returns:

Returns no value.

6.80.5.61 `unsigned int ewl_widget_onscreen_is (Ewl_Widget * w)`

Checks if the given widget is currently on screen.

Parameters:

w,: The widget to check

Returns:

Returns TRUE if the widget is onscreen

6.80.5.62 Ewl_Widget* ewl_widget_parent_get (Ewl_Widget * *w*)

Retrieves the parent of the given widget.

Parameters:

w,: The widget to get the parent from

Returns:

Returns the parent of the given widget, or NULL if none set

6.80.5.63 int ewl_widget_parent_of (Ewl_Widget * *c*, Ewl_Widget * *w*)

Determine if a widget is a parent of another widget.

Parameters:

c,: the widget to compare against

w,: the widget to check parentage

Returns:

Returns TRUE if *c* is a parent of *w*, otherwise returns FALSE.

6.80.5.64 void ewl_widget_parent_set (Ewl_Widget * *w*, Ewl_Widget * *p*)

change the parent of the specified widget

Parameters:

w,: the widget to change the parent

p,: the new parent of the widget

Returns:

Returns no value.

Changes the parent of the widget *w*, to the container *p*. The reparent callback is triggered to notify children of *w* of the change in parent.

6.80.5.65 void ewl_widget_print (Ewl_Widget * *w*)

Prints info for debugging a widget's state information.

Parameters:

w,: the widget to print info

Returns:

Returns no value.

6.80.5.66 void ewl_widget_print_verbose (Ewl_Widget * *w*)

Prints verbose info for debugging a widget's state information.

Parameters:

w,: the widget to print verbose info

Returns:

Returns no value.

6.80.5.67 void ewl_widget_realize (Ewl_Widget * *w*)

Realize the specified widget.

Parameters:

w,: the widget to realize

Returns:

Returns no value.

The specified widget is realized, ie. actually displayed to the screen.

6.80.5.68 void ewl_widget_reparent (Ewl_Widget * *w*)

initiate reparent of the specified widget

Parameters:

w,: the widget to reparent

Returns:

Returns no value.

The reparent callback is triggered for the specified widget, this should adjust the widgets attributes based on the new parent.

6.80.5.69 void ewl_widget_reveal (Ewl_Widget * *w*)

Indicate a widget is revealed.

Parameters:

w,: the widget to mark as revealed

Returns:

Returns no value.

6.80.5.70 void ewl_widget_show (Ewl_Widget * *w*)

mark a widget as visible

Parameters:

w,: the widget to be marked as visible

Returns:

Returns no value.

Marks the widget as visible so that it will be displayed the next time through the rendering loop. Note that the show callback may be delayed until the widget has been realized.

6.80.5.71 void ewl_widget_state_set (Ewl_Widget * *w*, const char * *state*, Ewl_State_Type *flag*)

Update the appearance of the widget to a state.

Parameters:

w,: the widget to update the appearance

state,: the new state of the widget

flag,: the flag for the state e.g. EWL_STATE_TRANSIENT

Returns:

Returns no value.

Changes the appearance of the widget depending on the state string passed by the state parameter.

6.80.5.72 void ewl_widget_tab_order_append (Ewl_Widget * *w*)

Changes the order in the embed so *w* receives focus first on tab.

Parameters:

w,: the widget to be moved to the front of the focus list

Returns:

Returns no value.

This moves the widget *w* to the end of the tab order list in the embed that holds it.

6.80.5.73 void ewl_widget_tab_order_insert (Ewl_Widget * *w*, unsigned int *idx*)

Changes the order in the embed so *w* receives focus first on tab.

Parameters:

w,: the widget to be moved to the front of the focus list

idx,: The index to insert the tab into

Returns:

Returns no value.

This moves the widget *w* to the given index in the tab order list in the embed that holds it.

6.80.5.74 void ewl_widget_tab_order_insert_after (Ewl_Widget * *w*,
Ewl_Widget * *after*)

Insert the given widget into the tab order after the *after* widget.

Parameters:

w,: The widget to insert into the tab order

after,: The widget to insert after

Returns:

Returns no value.

6.80.5.75 void ewl_widget_tab_order_insert_before (Ewl_Widget * *w*,
Ewl_Widget * *before*)

Inserts the widget into the tab order before the *before* widget.

Parameters:

w,: The widget to be inserted into the tab order

before,: The widget we are to be inserted before

Returns:

Returns no value.

6.80.5.76 void ewl_widget_tab_order_prepend (Ewl_Widget * *w*)

Changes the order in the embed so *w* receives focus first on tab.

Parameters:

w,: the widget to be moved to the front of the focus list

Returns:

Returns no value.

This moves the widget *w* to the front of the tab order list in the embed that holds it.

6.80.5.77 void ewl_widget_tab_order_remove (Ewl_Widget * *w*)

Remove the widget from the tab order.

Parameters:

w,: The widget to remove from the tab order

Returns:

Returns no value.

6.80.5.78 void ewl_widget_tree_print (Ewl_Widget * *w*)

Prints to stdout the tree of widgets that are parents of a widget.

Parameters:

w,: the widget to display ancestry tree

Returns:

Returns no value.

6.80.5.79 unsigned int ewl_widget_type_is (Ewl_Widget * *widget*, const char * *type*)

Determine if the widget *w* has inherited from the type *t*.

Parameters:

widget,: the widget to determine if a type is inherited

type,: the type to check for inheritance in the widget

Returns:

Returns TRUE if *w* inherited the type *t*, otherwise FALSE.

6.80.5.80 void ewl_widget_unrealize (Ewl_Widget * *w*)

Unrealize the specified widget.

Parameters:

w,: the widget to unrealize

Returns:

Returns no value.

The specified widget is unrealized, ie. no longer displayed to the screen.

6.81 Ewl_Window: A Container for Displaying in a New Window

6.81.1 Detailed Description

Defines the **Ewl_Window** (p. 788) class which extends the **Ewl_Embed** (p. 629) class by creating it's own window and evas.

Remarks:

Inherits from **Ewl_Embed** (p. 629).

Data Structures

- struct **Ewl_Window**

*Inherits from **Ewl_Embed** (p. 629) class to create it's own window and canvas for drawing, sizing and positioning.*

Defines

- #define **EWL_WINDOW**(win) ((**Ewl_Window** *) win)
- #define **EWL_WINDOW_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_WINDOW_TYPE**))
- #define **EWL_WINDOW_TYPE** "window"

Typedefs

- typedef **Ewl_Window** **Ewl_Window**

Functions

- void **ewl_window_attention_demand** (**Ewl_Window** *win)
Request the WM to pay attention to the window.
- void **ewl_window_borderless_set** (**Ewl_Window** *win)
Remove the border from the specified window.
- void **ewl_window_cb_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_window_cb_destroy** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_window_cb_expose** (**Ewl_Widget** *w, void *ev, void *user_data)
- void **ewl_window_cb_hide** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_window_cb_postrealize** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_window_cb_realize** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_window_cb_realize_transient** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_window_cb_show** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_window_cb_unrealize** (**Ewl_Widget** *w, void *ev_data, void *user_data)

- `char * ewl_window_class_get (Ewl_Window *win)`
Retrieve the class of the specified window.
- `void ewl_window_class_set (Ewl_Window *win, const char *classname)`
Set the class of the specified window.
- `int ewl_window_dialog_get (Ewl_Window *win)`
Retrieves the current dialog state on a window.
- `void ewl_window_dialog_set (Ewl_Window *win, int dialog)`
Changes the dialog state on the specified window.
- `unsigned int ewl_window_fullscreen_get (Ewl_Window *win)`
Retrieve the fullscreen setting for the window.
- `void ewl_window_fullscreen_set (Ewl_Window *win, unsigned int fullscreen)`
Sets the fullscreen setting for the window.
- `int ewl_window_init (Ewl_Window *win)`
Initialize a window to default values and callbacks.
- `int ewl_window_keyboard_grab_get (Ewl_Window *win)`
Retrieves the current keyboard grab state on a window.
- `void ewl_window_keyboard_grab_set (Ewl_Window *win, int grab)`
Changes the keyboard grab state on the specified window.
- `void ewl_window_lower (Ewl_Window *win)`
Lower a window.
- `int ewl_window_modal_get (Ewl_Window *win)`
Gets the boolean flag indicating if win is modal.
- `void ewl_window_modal_set (Ewl_Window *win, int modal)`
Sets the window to modal or non-modal based on modal.
- `void ewl_window_move (Ewl_Window *win, int x, int y)`
Move the specified window to the given position.
- `char * ewl_window_name_get (Ewl_Window *win)`
Retrieve the name of the specified window.
- `void ewl_window_name_set (Ewl_Window *win, const char *name)`
Set the name of the specified window.
- `Ewl_Widget * ewl_window_new (void)`
Allocate and initialize a new window.
- `int ewl_window_override_get (Ewl_Window *win)`
Retrieves the current override state on a window.

- **void ewl_window_override_set** (Ewl_Window *win, int override)
Changes the override state on the specified window.
- **int ewl_window_pointer_grab_get** (Ewl_Window *win)
Retrieves the current pointer grab state on a window.
- **void ewl_window_pointer_grab_set** (Ewl_Window *win, int grab)
Changes the pointer grab state on the specified window.
- **void ewl_window_raise** (Ewl_Window *win)
Raise a window.
- **void ewl_window_selection_text_set** (Ewl_Window *win, const char *txt)
This will set the given txt as the selection text on the window or clear the text if txt is NULL.
- **unsigned int ewl_window_skip_pager_get** (Ewl_Window *win)
Retrieve the skip pager setting for the window.
- **void ewl_window_skip_pager_set** (Ewl_Window *win, unsigned int skip)
Sets the skip pager setting for the window.
- **unsigned int ewl_window_skip_taskbar_get** (Ewl_Window *win)
Retrieve the skip taskbar setting for the window.
- **void ewl_window_skip_taskbar_set** (Ewl_Window *win, unsigned int skip)
Sets the skip taskbar setting for the window.
- **char * ewl_window_title_get** (Ewl_Window *win)
Retrieve the title of the specified window.
- **void ewl_window_title_set** (Ewl_Window *win, const char *title)
Set the title of the specified window.
- **void ewl_window_transient_for** (Ewl_Window *win, Ewl_Window *forwin)
Sets a window to be transient for another window.
- **void ewl_window_transient_for_foreign** (Ewl_Window *win, Ewl_Embed_Window *forwin)
Sets a window to be transient for another window.
- **Ewl_Window * ewl_window_window_find** (void *window)
Find an ewl window by it's X window.

Variables

- **int EWL_CALLBACK_DELETE_WINDOW**
- **int EWL_CALLBACK_EXPOSE**

6.81.2 Define Documentation

6.81.2.1 `#define EWL_WINDOW(win) ((Ewl_Window *) win)`

Typecasts a pointer to an **Ewl_Window** (p. 788) pointer.

6.81.2.2 `#define EWL_WINDOW_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_WINDOW_TYPE))`

Returns TRUE if the widget is an **Ewl_Window** (p. 788), FALSE otherwise

6.81.2.3 `#define EWL_WINDOW_TYPE "window"`

The type name for the **Ewl_Window** (p. 788) widget

6.81.3 Typedef Documentation

6.81.3.1 `typedef struct Ewl_Window Ewl_Window`

The window structure is mostly a container for holding widgets and a wrapper around the xlib window.

6.81.4 Function Documentation

6.81.4.1 `void ewl_window_attention_demand (Ewl_Window * win)`

Request the WM to pay attention to the window.

Parameters:

win,: the window to work with.

Returns:

Returns no value.

Demand attention for the window *win* if it is realized.

6.81.4.2 `void ewl_window_borderless_set (Ewl_Window * win)`

Remove the border from the specified window.

Parameters:

win,: the window to remove the border

Returns:

Returns no value.

Remove the border from the specified widget and call the necessary X lib functions to update the appearance.

- 6.81.4.3 void ewl_window_cb_configure (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.81.4.4 void ewl_window_cb_destroy (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.81.4.5 void ewl_window_cb_expose (Ewl_Widget * *w*, void * *ev*, void * *user_data*)
- 6.81.4.6 void ewl_window_cb_hide (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.81.4.7 void ewl_window_cb_postrealize (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.81.4.8 void ewl_window_cb_realize (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.81.4.9 void ewl_window_cb_realize_transient (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.81.4.10 void ewl_window_cb_show (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.81.4.11 void ewl_window_cb_unrealize (Ewl_Widget * *w*, void * *ev_data*, void * *user_data*)
- 6.81.4.12 char* ewl_window_class_get (Ewl_Window * *win*)

Retrieve the class of the specified window.

Parameters:

win,: the window to retrieve the window

Returns:

Returns a pointer to a new copy of the class, NULL on failure.

The returned class should be freed.

- 6.81.4.13 void ewl_window_class_set (Ewl_Window * *win*, const char * *classname*)

Set the class of the specified window.

Parameters:

win,: the window to change the class

classname,: the class to set for the window

Returns:

Returns no value.

Sets the class of window *w* to *class* and calls the necessary X lib functions to update the window.

6.81.4.14 `int ewl_window_dialog_get (Ewl_Window * win)`

Retrieves the current dialog state on a window.

Parameters:

win,: window to retrieve dialog state

Returns:

Returns TRUE if window is an dialog window, FALSE otherwise.

6.81.4.15 `void ewl_window_dialog_set (Ewl_Window * win, int dialog)`

Changes the dialog state on the specified window.

Parameters:

win,: the window

dialog,: TRUE or FALSE to indicate dialog state.

Returns:

Returns no value.

A dialog window has not a iconify and/or maximize button.

6.81.4.16 `unsigned int ewl_window_fullscreen_get (Ewl_Window * win)`

Retrieve the fullscreen setting for the window.

Parameters:

win,: The window to work with

Returns:

Returns TRUE if the window is fullscreen, FALSE otherwise

6.81.4.17 `void ewl_window_fullscreen_set (Ewl_Window * win, unsigned int fullscreen)`

Sets the fullscreen setting for the window.

Parameters:

win,: The window to work with

fullscreen,: The fullscreen setting to use

Returns:

Returns no value

6.81.4.18 int ewl_window_init (Ewl_Window * *w*)

Initialize a window to default values and callbacks.

Parameters:

w,: the window to be initialized to default values and callbacks

Returns:

Returns TRUE or FALSE depending on if initialization succeeds.

Sets the values and callbacks of a window *w* to their defaults.

6.81.4.19 int ewl_window_keyboard_grab_get (Ewl_Window * *win*)

Retrieves the current keyboard grab state on a window.

Parameters:

win,: window to retrieve keyboard grab state

Returns:

Returns TRUE if window is grabbing keyboard, FALSE otherwise.

6.81.4.20 void ewl_window_keyboard_grab_set (Ewl_Window * *win*, int *grab*)

Changes the keyboard grab state on the specified window.

Parameters:

win,: the window to change keyboard grab settings.

grab,: TRUE or FALSE to indicate grab state.

Returns:

Returns no value.

6.81.4.21 void ewl_window_lower (Ewl_Window * *win*)

Lower a window.

Parameters:

win,: the window to lower.

Returns:

Returns no value.

Lower the window *win* if it is realized.

6.81.4.22 `int ewl_window_modal_get (Ewl_Window * win)`

Gets the boolean flag indicating if *win* is modal.

Parameters:

win,: The window to work with

Returns:

Returns a boolean indicating if the window is modal.

6.81.4.23 `void ewl_window_modal_set (Ewl_Window * win, int modal)`

Sets the window to modal or non-modal based on *modal*.

Parameters:

win,: The window to work with

modal,: Boolean to indicate if this window is modal

Returns:

Returns no value

6.81.4.24 `void ewl_window_move (Ewl_Window * win, int x, int y)`

Move the specified window to the given position.

Parameters:

win,: the window to move

x,: the x coordinate of the new position

y,: the y coordinate of the new position

Returns:

Returns no value.

Moves the window into the specified position in the window manager environment.

6.81.4.25 `char* ewl_window_name_get (Ewl_Window * win)`

Retrieve the name of the specified window.

Parameters:

win,: the window to retrieve the window

Returns:

Returns a pointer to a new copy of the name, NULL on failure.

The returned name should be freed.

6.81.4.26 void ewl_window_name_set (Ewl_Window * *win*, const char * *name*)

Set the name of the specified window.

Parameters:

win,: the window to change the name

name,: the name to set for the window

Returns:

Returns no value.

Sets the name of window *w* to *name* and calls the necessary X lib functions to update the window.

6.81.4.27 Ewl_Widget* ewl_window_new (void)

Allocate and initialize a new window.

Returns:

Returns a new window on success, or NULL on failure.

6.81.4.28 int ewl_window_override_get (Ewl_Window * *win*)

Retrieves the current override state on a window.

Parameters:

win,: window to retrieve override state

Returns:

Returns TRUE if window is an override window, FALSE otherwise.

6.81.4.29 void ewl_window_override_set (Ewl_Window * *win*, int *override*)

Changes the override state on the specified window.

Parameters:

win,: the window to change override settings

override,: TRUE or FALSE to indicate override state.

Returns:

Returns no value.

6.81.4.30 `int ewl_window_pointer_grab_get (Ewl_Window * win)`

Retrieves the current pointer grab state on a window.

Parameters:

win,: window to retrieve pointer grab state

Returns:

Returns TRUE if window is grabbing pointer, FALSE otherwise.

6.81.4.31 `void ewl_window_pointer_grab_set (Ewl_Window * win, int grab)`

Changes the pointer grab state on the specified window.

Parameters:

win,: the window to change pointer grab settings.

grab,: TRUE or FALSE to indicate grab state.

Returns:

Returns no value.

6.81.4.32 `void ewl_window_raise (Ewl_Window * win)`

Raise a window.

Parameters:

win,: the window to raise.

Returns:

Returns no value.

Raise the window *win* if it is realized.

6.81.4.33 `void ewl_window_selection_text_set (Ewl_Window * win, const char * txt)`

This will set the given *txt* as the selection text on the window or clear the text if *txt* is NULL.

Parameters:

win,: The window to set the selection on

txt,: The text to set into the selection

Returns:

Returns no value.

,

6.81.4.34 unsigned int ewl_window_skip_pager_get (Ewl_Window * win)

Retrieve the skip pager setting for the window.

Parameters:

win,: The window to work with

Returns:

Returns TRUE if the window is to be skipped for the pager, FALSE otherwise

6.81.4.35 void ewl_window_skip_pager_set (Ewl_Window * win, unsigned int skip)

Sets the skip pager setting for the window.

Parameters:

win,: The window to work with

skip,: If the win should be visible in the pager

Returns:

Returns no value

6.81.4.36 unsigned int ewl_window_skip_taskbar_get (Ewl_Window * win)

Retrieve the skip taskbar setting for the window.

Parameters:

win,: The window to work with

Returns:

Returns TRUE if the window is to be skipped for the taskbar, FALSE otherwise

6.81.4.37 void ewl_window_skip_taskbar_set (Ewl_Window * win, unsigned int skip)

Sets the skip taskbar setting for the window.

Parameters:

win,: The window to work with

skip,: If the win should not be in the taskbar

Returns:

Returns no value

6.81.4.38 `char* ewl_window_title_get (Ewl_Window * win)`

Retrieve the title of the specified window.

Parameters:

win,: the window to retrieve the window

Returns:

Returns a pointer to a new copy of the title, NULL on failure.

The returned title should be freed.

6.81.4.39 `void ewl_window_title_set (Ewl_Window * win, const char * title)`

Set the title of the specified window.

Parameters:

win,: the window to change the title

title,: the title to set for the window

Returns:

Returns no value.

Sets the title of window *w* to *title* and calls the necessary X lib functions to update the window.

6.81.4.40 `void ewl_window_transient_for (Ewl_Window * win, Ewl_Window * forwin)`

Sets a window to be transient for another window.

Parameters:

win,: window to set transient

forwin,: the window to be transient for

Returns:

Returns no value.

6.81.4.41 `void ewl_window_transient_for_foreign (Ewl_Window * win,
Ewl_Embed_Window * forwin)`

Sets a window to be transient for another window.

Parameters:

win,: window to set transient

forwin,: the window to be transient for

Returns:

Returns no value.

6.81.4.42 Ewl_Window* ewl_window_window_find (void * *window*)

Find an ewl window by it's X window.

Parameters:

window,: the X window to search for on the list of ewl window's

Returns:

Returns the found ewl window on success, NULL on failure.

6.81.5 Variable Documentation**6.81.5.1 int EWL_CALLBACK_DELETE_WINDOW**

The window is being closed

6.81.5.2 int EWL_CALLBACK_EXPOSE

Triggered when the window needs redrawing

Chapter 7

Enlightened Widget Library Data Structure Documentation

7.1 Combo_Test_Data Struct Reference

Data Fields

- unsigned int **count**
- char ** **data**

7.1.1 Field Documentation

7.1.1.1 unsigned int Combo_Test_Data::count

7.1.1.2 char Combo_Test_Data::data**

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

- bin/tests/combo/**ewl_combo.c**

7.2 Control Struct Reference

Data Fields

- `void(* cb)(Ewl_Widget *w, void *ev, void *data)`
- `Ewl_Callback_Function func`
- `char * name`
- `Ewl_Stock_Type name`

7.2.1 Field Documentation

7.2.1.1 `void(* Control::cb)(Ewl_Widget *w, void *ev, void *data)`

7.2.1.2 `Ewl_Callback_Function Control::func`

7.2.1.3 `char* Control::name`

7.2.1.4 `Ewl_Stock_Type Control::name`

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

- `bin/tests/media/ewl_media.c`
- `bin/tests/text_editor/ewl_text_editor.c`

7.3 Ewl_Attach Struct Reference

```
#include <ewl_attach.h>
```

7.3.1 Detailed Description

Contains information about widget attachments.

Data Fields

- `void * data`
- `Ewl_Attach_Data_Type data_type`
- `void * priv_data`
- `Ewl_Attach_Type type`

7.3.2 Field Documentation

7.3.2.1 `void* Ewl_Attach::data`

The attachment data

7.3.2.2 `Ewl_Attach_Data_Type Ewl_Attach::data_type`

The data type of the attachment

7.3.2.3 `void* Ewl_Attach::priv_data`

Any private data in the attachment

7.3.2.4 `Ewl_Attach_Type Ewl_Attach::type`

The type of the attachment

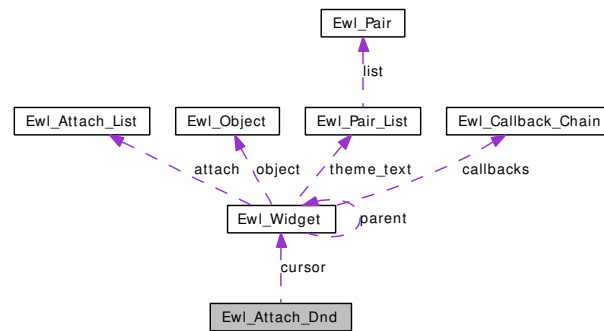
The documentation for this struct was generated from the following file:

- `lib/ewl_attach.h`

7.4 Ewl_Attach_Dnd Struct Reference

```
#include <ewl_attach.h>
```

Collaboration diagram for Ewl_Attach_Dnd:



7.4.1 Detailed Description

DND data associated with the widget.

Data Fields

- `Ewl_Widget * cursor`
- `void * data`
- `int size`

7.4.2 Field Documentation

7.4.2.1 `Ewl_Widget* Ewl_Attach_Dnd::cursor`

Cursor displayed for DND

7.4.2.2 `void* Ewl_Attach_Dnd::data`

The attachment data

7.4.2.3 `int Ewl_Attach_Dnd::size`

Any private data in the attachment

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

- `lib/ewl_attach.h`

7.5 Ewl_Attach_List Struct Reference

```
#include <ewl_widget.h>
```

7.5.1 Detailed Description

A list of things attached to a widget.

Data Fields

- unsigned int **direct**:1
- unsigned int **len**:31
- void ** **list**

7.5.2 Field Documentation

7.5.2.1 unsigned int Ewl_Attach_List::direct

Is this a direct pointer to an attached item

7.5.2.2 unsigned int Ewl_Attach_List::len

Number of things in the attachment list

7.5.2.3 void** Ewl_Attach_List::list

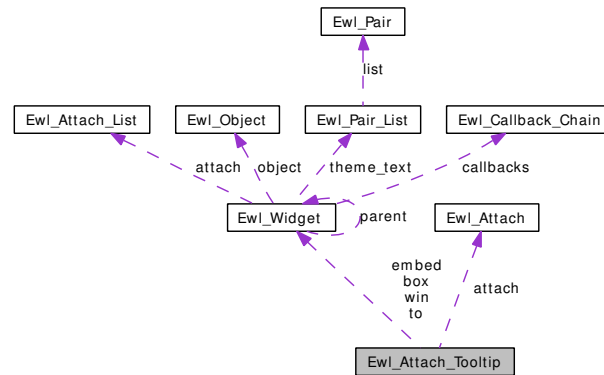
The attachment list

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

- lib/ewl_widget.h

7.6 Ewl_Attach_Tooltip Struct Reference

Collaboration diagram for Ewl_Attach_Tooltip:



7.6.1 Detailed Description

Inherits from **Ewl_Widget** (p. 785) and extends to provide information on a tooltip.

Data Fields

- **Ewl_Attach * attach**
- **Ewl_Widget * box**
- **Ewl_Widget * embed**
- **Ecore_Timer * timer**
- **Ewl_Widget * to**
- **Ewl_Widget * win**
- **Evas_Coord x**
- **Evas_Coord y**

7.6.2 Field Documentation

7.6.2.1 Ewl_Attach* Ewl_Attach_Tooltip::attach

7.6.2.2 Ewl_Widget* Ewl_Attach_Tooltip::box

7.6.2.3 Ewl_Widget* Ewl_Attach_Tooltip::embed

7.6.2.4 Ecore_Timer* Ewl_Attach_Tooltip::timer

7.6.2.5 Ewl_Widget* Ewl_Attach_Tooltip::to

7.6.2.6 Ewl_Widget* Ewl_Attach_Tooltip::win

7.6.2.7 Evas_Coord Ewl_Attach_Tooltip::x

7.6.2.8 Evas_Coord Ewl_Attach_Tooltip::y

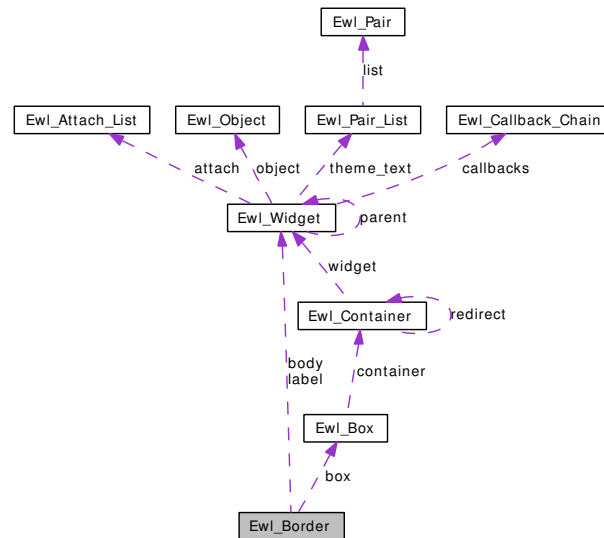
The documentation for this struct was generated from the following file:

- lib/ewl_attach.c

7.7 Ewl_Border Struct Reference

```
#include <ewl_border.h>
```

Collaboration diagram for Ewl_Border:



7.7.1 Detailed Description

Inherits from **Ewl_Box** (p. 586) to allow drawing a border and label decoration around widgets.

Data Fields

- **Ewl_Widget * body**
- **Ewl_Box box**
- **Ewl_Widget * label**
- **Ewl_Position label_position**

7.7.2 Field Documentation

7.7.2.1 Ewl_Widget* Ewl_Border::body

Box for holding children

7.7.2.2 Ewl_Box Ewl_Border::box

Inherit from **Ewl_Box** (p. 586)

7.7.2.3 Ewl_Widget* Ewl_Border::label

Text label for the border

7.7.2.4 Ewl_Position Ewl_Border::label_position

Flags for placing the label

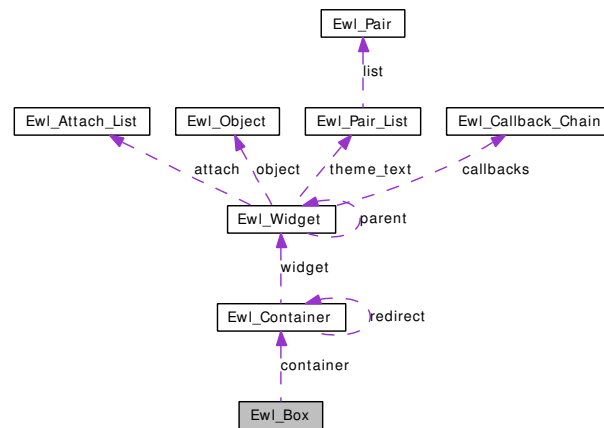
The documentation for this struct was generated from the following file:

- lib/ewl_border.h

7.8 Ewl_Box Struct Reference

```
#include <ewl_box.h>
```

Collaboration diagram for Ewl_Box:



7.8.1 Detailed Description

Inherits from an **Ewl_Container** (p. 618) to provide layout facilities for child widgets placed inside.

Layout is done horizontally (left-to-right), or vertically (top-to-bottom). There is also a flag for homogeneous layout, which gives each child **Ewl_Widget** (p. 785) equal space inside the **Ewl_Box** (p. 586).

Data Fields

- **Ewl_Container** container
- unsigned int **homogeneous**
- **Ewl_Orientation** orientation
- int **spacing**

7.8.2 Field Documentation

7.8.2.1 Ewl_Container Ewl_Box::container

Inherit from **Ewl_Container** (p. 618)

7.8.2.2 unsigned int Ewl_Box::homogeneous

Flag indicating space assignemnt

7.8.2.3 Ewl_Orientation Ewl_Box::orientation

Indicate the orientation of layout

7.8.2.4 int Ewl_Box::spacing

Space between each widget in the box

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

- lib/ewl_box.h

7.9 Ewl_Box_Orientation Struct Reference

7.9.1 Detailed Description

Provides information on the orientation of a box.

Data Fields

- `int a1_align`
- `int a3_align`
- `int(* align_ask)(Ewl_Object *ob)`
- `void(* align_set)(Ewl_Object *ob, int size)`
- `int f1_align`
- `int f3_align`
- `int f_policy`
- `int(* fill_ask)(Ewl_Object *ob)`
- `void(* fill_set)(Ewl_Object *ob, int size)`
- `void(* pref_fill_set)(Ewl_Object *ob, int size)`

7.9.2 Field Documentation

7.9.2.1 `int Ewl_Box_Orientation::a1_align`

7.9.2.2 `int Ewl_Box_Orientation::a3_align`

7.9.2.3 `int(* Ewl_Box_Orientation::align_ask)(Ewl_Object *ob)`

7.9.2.4 `void(* Ewl_Box_Orientation::align_set)(Ewl_Object *ob, int size)`

7.9.2.5 `int Ewl_Box_Orientation::f1_align`

7.9.2.6 `int Ewl_Box_Orientation::f3_align`

7.9.2.7 `int Ewl_Box_Orientation::f_policy`

7.9.2.8 `int(* Ewl_Box_Orientation::fill_ask)(Ewl_Object *ob)`

7.9.2.9 `void(* Ewl_Box_Orientation::fill_set)(Ewl_Object *ob, int size)`

7.9.2.10 `void(* Ewl_Box_Orientation::pref_fill_set)(Ewl_Object *ob, int size)`

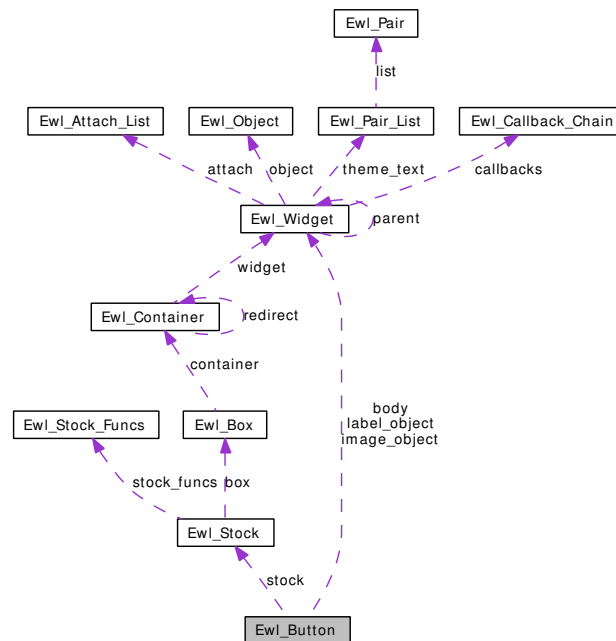
The documentation for this struct was generated from the following file:

- `lib/ewl_box.c`

7.10 Ewl_Button Struct Reference

```
#include <ewl_button.h>
```

Collaboration diagram for Ewl_Button:



7.10.1 Detailed Description

A simple **Ewl_Widget** (p. 785) to provide for a clickable button in the UI.

Provides easy facilities for adding a **Ewl_Label** (p. 692) label to the button, and a **Ewl_Image** (p. 686) but allows for placing any number of **Ewl_Widget**'s in the **Ewl_Button** (p. 589).

Data Fields

- **Ewl_Widget * body**
- **Ewl_Widget * image_object**
- **Ewl_Widget * label_object**
- **Ewl_Stock stock**

7.10.2 Field Documentation

7.10.2.1 Ewl_Widget* Ewl_Button::body

The body of the button

7.10.2.2 Ewl_Widget* Ewl_Button::image_object

Add an image to the button if needed

7.10.2.3 Ewl_Widget* Ewl_Button::label_object

Labels are common, make it easy

7.10.2.4 Ewl_Stock Ewl_Button::stock

Inherit from the box for adding widgets

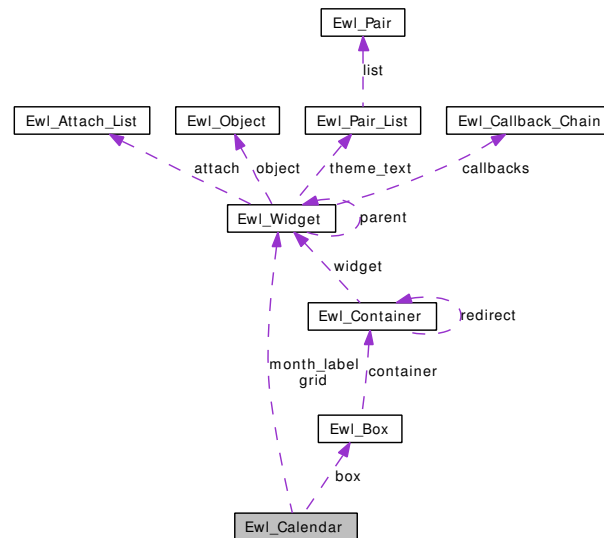
The documentation for this struct was generated from the following file:

- lib/ewl_button.h

7.11 Ewl_Calendar Struct Reference

```
#include <ewl_calendar.h>
```

Collaboration diagram for Ewl_Calendar:



7.11.1 Detailed Description

Inherits from an **Ewl_Box** (p. 586) to provide a calendar widget.

Layout is either free-form, or auto-arranged to a grid.

Data Fields

- **Ewl_Box** box
- int cur_day
- int cur_month
- int cur_year
- **Ewl_Widget** * grid
- **Ewl_Widget** * month_label

7.11.2 Field Documentation

7.11.2.1 Ewl_Box Ewl_Calendar::box

Inherit from **Ewl_Box** (p. 586)

7.11.2.2 int Ewl_Calendar::cur_day

The current day selected

7.11.2.3 int Ewl_Calendar::cur_month

The current month selected

7.11.2.4 int Ewl_Calendar::cur_year

The current year selected

7.11.2.5 Ewl_Widget* Ewl_Calendar::grid

The layout grid

7.11.2.6 Ewl_Widget* Ewl_Calendar::month_label

The month label

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

- lib/ewl_calendar.h

7.12 Ewl_Callback Struct Reference

```
#include <ewl_callback.h>
```

7.12.1 Detailed Description

Contains information about a callback.

Data Fields

- **Ewl_Callback_Function func**
- **int id**
- **int references**
- **void * user_data**

7.12.2 Field Documentation

7.12.2.1 Ewl_Callback_Function Ewl_Callback::func

Function executed

7.12.2.2 int Ewl_Callback::id

id number of this callback

7.12.2.3 int Ewl_Callback::references

Reference counting

7.12.2.4 void* Ewl_Callback::user_data

user specified data to pass to func

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

- **lib/ewl_callback.h**

7.13 Ewl_Callback_Chain Struct Reference

```
#include <ewl_widget.h>
```

7.13.1 Detailed Description

The callback chain contains the length, mask and information on the list.

Data Fields

- unsigned char **index**
- unsigned char **len**
- void ** **list**
- unsigned char **mask**

7.13.2 Field Documentation

7.13.2.1 unsigned char Ewl_Callback_Chain::index

Current list index

7.13.2.2 unsigned char Ewl_Callback_Chain::len

Length of the list

7.13.2.3 void** Ewl_Callback_Chain::list

The callback list

7.13.2.4 unsigned char Ewl_Callback_Chain::mask

The callback mask

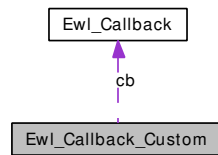
The documentation for this struct was generated from the following file:

- lib/**ewl_widget.h**

7.14 Ewl_Callback_Custom Struct Reference

```
#include <ewl_callback.h>
```

Collaboration diagram for Ewl_Callback_Custom:



7.14.1 Detailed Description

Contains information about custom callbacks.

Data Fields

- **Ewl_Callback cb**
- unsigned int **event_id**

7.14.2 Field Documentation

7.14.2.1 Ewl_Callback Ewl_Callback_Custom::cb

Inherit from **Ewl_Callback** (p. 593)

7.14.2.2 unsigned int Ewl_Callback_Custom::event_id

Add an event id

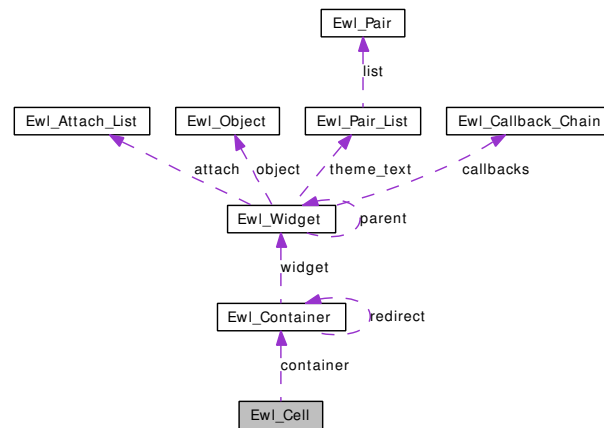
The documentation for this struct was generated from the following file:

- lib/**ewl_callback.h**

7.15 Ewl_Cell Struct Reference

```
#include <ewl_cell.h>
```

Collaboration diagram for Ewl_Cell:



7.15.1 Detailed Description

Inherits from **Ewl_Container** (p. 618) for packing widgets inside a cell.

Data Fields

- **Ewl_Container** container

7.15.2 Field Documentation

7.15.2.1 Ewl_Container Ewl_Cell::container

Inherit from **Ewl_Container** (p. 618)

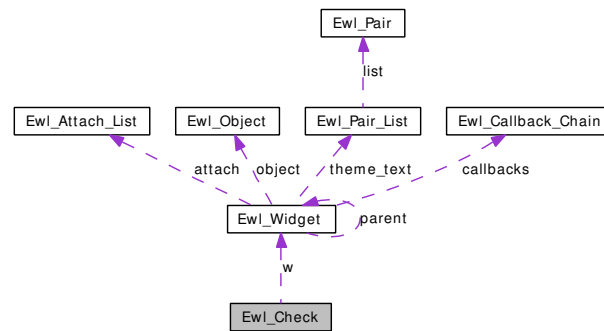
The documentation for this struct was generated from the following file:

- lib/ewl_cell.h

7.16 Ewl_Check Struct Reference

```
#include <ewl_check.h>
```

Collaboration diagram for Ewl_Check:



7.16.1 Detailed Description

Inherits from **Ewl_Widget** (p. 785) and extends to provide a checked state.

Data Fields

- `int checked`
- `Ewl_Widget w`

7.16.2 Field Documentation

7.16.2.1 `int Ewl_Check::checked`

Indicates if this is checked

7.16.2.2 `Ewl_Widget Ewl_Check::w`

Inherit from **Ewl_Widget** (p. 785)

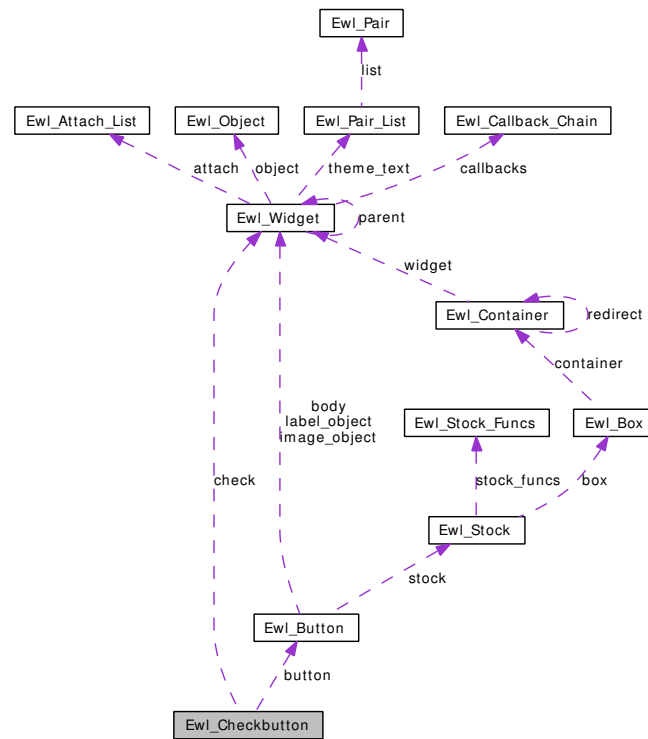
The documentation for this struct was generated from the following file:

- `lib/ewl_check.h`

7.17 Ewl_Checkbutton Struct Reference

```
#include <ewl_checkbutton.h>
```

Collaboration diagram for Ewl_Checkbutton:



7.17.1 Detailed Description

Inherits from **Ewl_Button** (p. 589) and expands to provide a stateful check button.

Data Fields

- **Ewl_Button** `button`
- **Ewl_Widget** * `check`
- **Ewl_Position** `label_position`

7.17.2 Field Documentation

7.17.2.1 Ewl_Button Ewl_Checkbutton::button

Inherit the basic button properties

7.17.2.2 Ewl_Widget* Ewl_Checkbutton::check

Check widget represented

7.17.2.3 Ewl_Position Ewl_Checkbutton::label_position

Order of label and check

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

- lib/ewl_checkbutton.h

7.18 Ewl_Color_Set Struct Reference

```
#include <ewl_widget.h>
```

7.18.1 Detailed Description

Contains an RGBA set of colours.

Data Fields

- int **a**
- int **b**
- int **g**
- int **r**

7.18.2 Field Documentation

7.18.2.1 int Ewl_Color_Set::a

The alpha value

7.18.2.2 int Ewl_Color_Set::b

The blue value

7.18.2.3 int Ewl_Color_Set::g

The green value

7.18.2.4 int Ewl_Color_Set::r

The red value

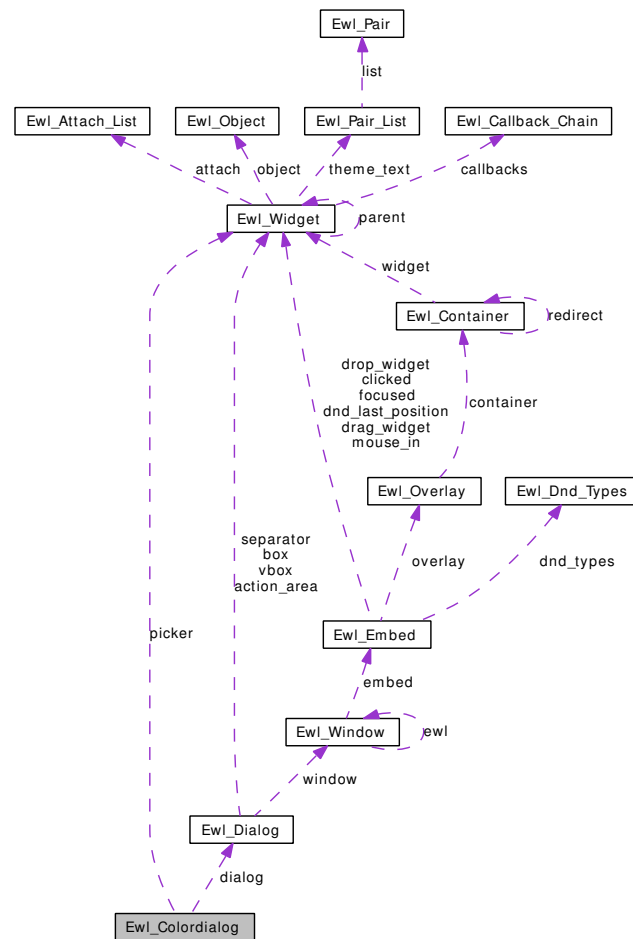
The documentation for this struct was generated from the following file:

- lib/**ewl_widget.h**

7.19 Ewl_Colordialog Struct Reference

```
#include <ewl_colordialog.h>
```

Collaboration diagram for Ewl_Colordialog:



7.19.1 Detailed Description

Inherit from the **Ewl_Dialog** (p. 626) to create a color dialog widget.

Data Fields

- **Ewl_Dialog** dialog
- **Ewl_Widget** * picker

7.19.2 Field Documentation

7.19.2.1 Ewl_Dialog Ewl_Colordialog::dialog

Inherits from **Ewl_Dialog** (p. 626)

7.19.2.2 Ewl_Widget* Ewl_Colordialog::picker

Store the color picker widget

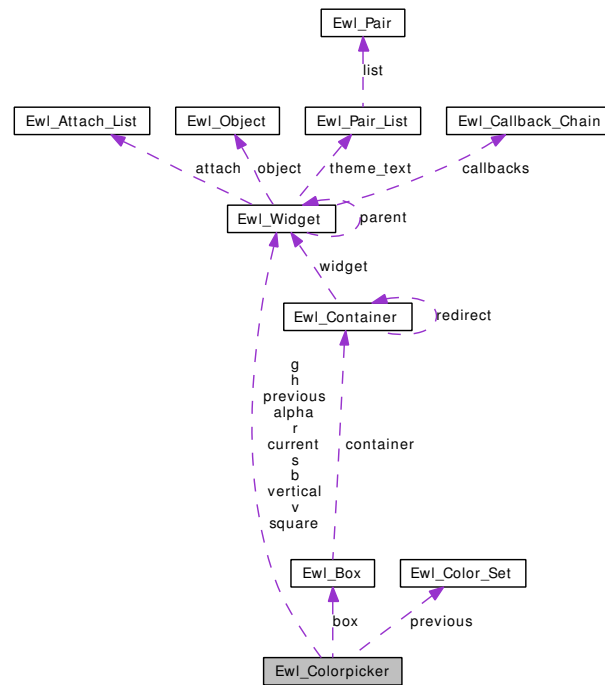
The documentation for this struct was generated from the following file:

- lib/ewl_colordialog.h

7.20 Ewl_Colorpicker Struct Reference

```
#include <ewl_colorpicker.h>
```

Collaboration diagram for Ewl_Colorpicker:



7.20.1 Detailed Description

Inherits from **Ewl_Box** (p. 586) to produce a color picker widget.

Data Fields

- **Ewl_Box** box
- unsigned int **has_alpha**:1
- **Ewl_Color_Mode** mode
- struct {
 - Ewl_Widget** * square
 - Ewl_Widget** * vertical
 } picker
- struct {
 - Ewl_Widget** * current
 - Ewl_Widget** * previous
 } preview
- **Ewl_Color_Set** previous
- struct {
 - Ewl_Widget** * alpha

```

    struct {
        Ewl_Widget * h
        Ewl_Widget * s
        Ewl_Widget * v
    } hsv
    struct {
        Ewl_Widget * b
        Ewl_Widget * g
        Ewl_Widget * r
    } rgb
} spinners

```

- unsigned int **updating**:1

7.20.2 Field Documentation

7.20.2.1 Ewl_Widget* Ewl_Colorpicker::alpha

The alpha value

7.20.2.2 Ewl_Widget* Ewl_Colorpicker::b

The blue value

7.20.2.3 Ewl_Box Ewl_Colorpicker::box

Inherit from **Ewl_Box** (p. 586)

7.20.2.4 Ewl_Widget* Ewl_Colorpicker::current

The current colour preview

7.20.2.5 Ewl_Widget* Ewl_Colorpicker::g

The green value

7.20.2.6 Ewl_Widget* Ewl_Colorpicker::h

The current hue

7.20.2.7 unsigned int Ewl_Colorpicker::has_alpha

Does the picker deal with alpha values

7.20.2.8 struct { ... } Ewl_Colorpicker::hsv

The pickers HSV values

7.20.2.9 Ewl_Color_Mode Ewl_Colorpicker::mode

The current mode

7.20.2.10 struct { ... } Ewl_Colorpicker::picker

This holds the two picker boxes

7.20.2.11 struct { ... } Ewl_Colorpicker::preview

The preview boxes

7.20.2.12 Ewl_Color_Set Ewl_Colorpicker::previous

The previous color

7.20.2.13 Ewl_Widget* Ewl_Colorpicker::previous

The previous colour preview

7.20.2.14 Ewl_Widget* Ewl_Colorpicker::r

The red value

7.20.2.15 struct { ... } Ewl_Colorpicker::rgb

The pickers RGB values

7.20.2.16 Ewl_Widget* Ewl_Colorpicker::s

The current saturation

7.20.2.17 struct { ... } Ewl_Colorpicker::spinners

The spinners to hold the different values

7.20.2.18 Ewl_Widget* Ewl_Colorpicker::square

The square picker box

7.20.2.19 unsigned int Ewl_Colorpicker::updating

Are we currently updating the widget

7.20.2.20 Ewl_Widget* Ewl_Colorpicker::v

The current value

7.20.2.21 Ewl_Widget* Ewl_Colorpicker::vertical

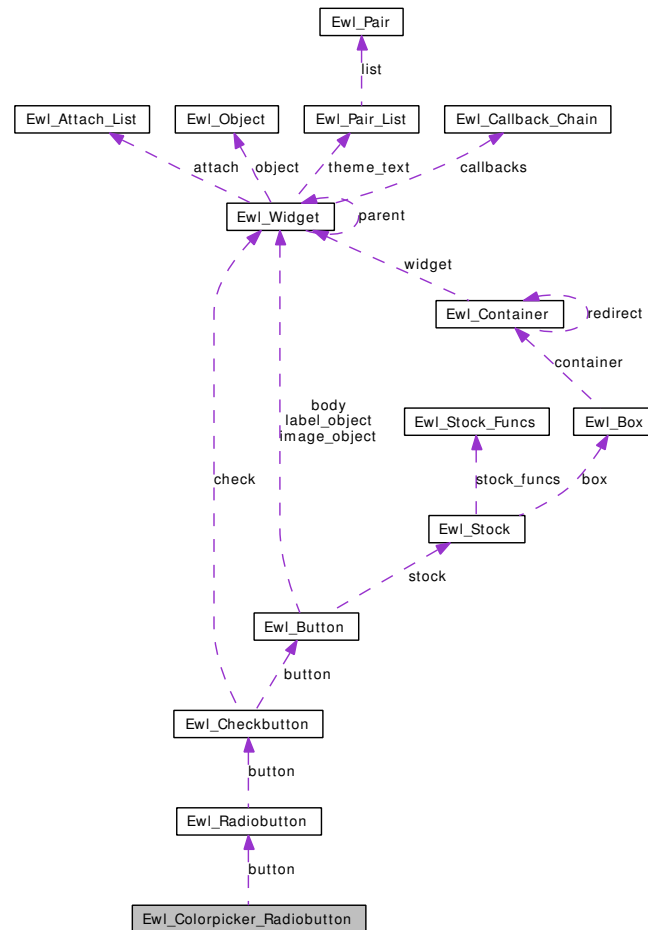
The vertical picker box

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

- lib/ewl_colorpicker.h

7.21 Ewl_Colorpicker_Radiobutton Struct Reference

Collaboration diagram for Ewl_Colorpicker_Radiobutton:



7.21.1 Detailed Description

Inherits from `Ewl_Radiobutton` (p. 725) and extends to provide colorpicker information.

Data Fields

- `Ewl_Radiobutton` `button`
- `Ewl_Color_Mode` `mode`

7.21.2 Field Documentation

7.21.2.1 `Ewl_Radiobutton` `Ewl_Colorpicker_Radiobutton::button`

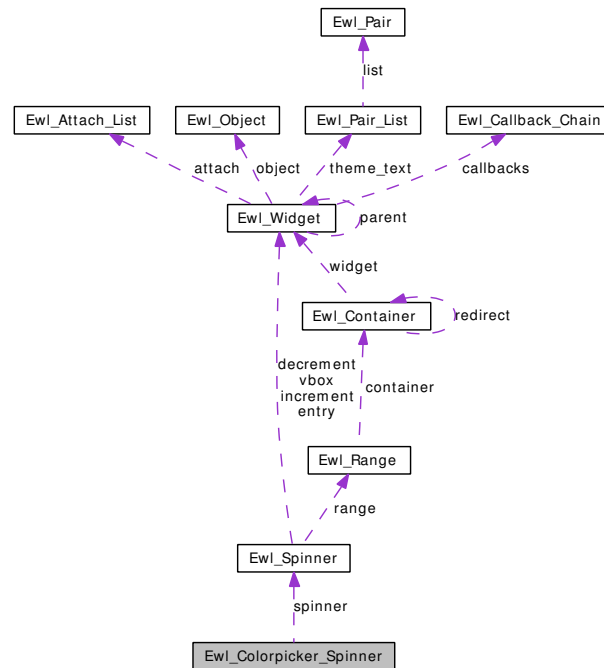
7.21.2.2 `Ewl_Color_Mode` `Ewl_Colorpicker_Radiobutton::mode`

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

- `lib/ewl_colorpicker.c`

7.22 Ewl_Colorpicker_Spinner Struct Reference

Collaboration diagram for Ewl_Colorpicker_Spinner:



7.22.1 Detailed Description

Inherits from `Ewl_Spinner` (p. 746) and extends to provide colorpicker information.

Data Fields

- `Ewl_Color_Mode` mode
- `Ewl_Spinner` spinner

7.22.2 Field Documentation

7.22.2.1 `Ewl_Color_Mode` `Ewl_Colorpicker_Spinner::mode`

7.22.2.2 `Ewl_Spinner` `Ewl_Colorpicker_Spinner::spinner`

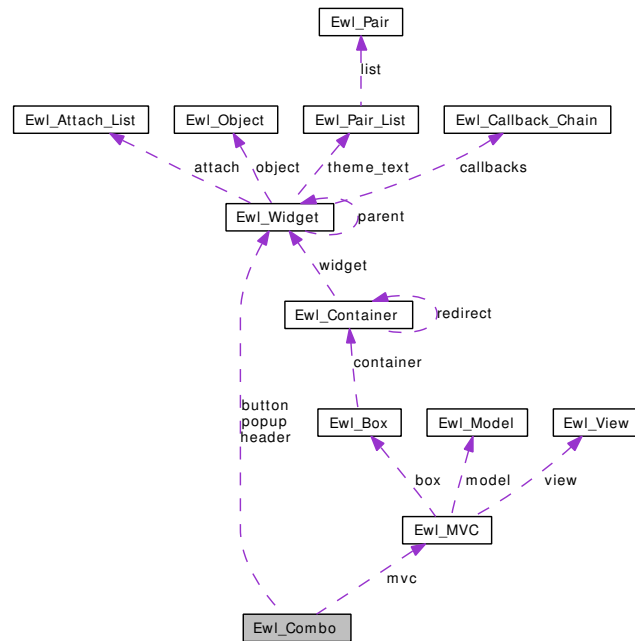
The documentation for this struct was generated from the following file:

- `lib/ewl_colorpicker.c`

7.23 Ewl_Combo Struct Reference

```
#include <ewl_combo.h>
```

Collaboration diagram for Ewl_Combo:



7.23.1 Detailed Description

Inherits from the **Ewl_MVC** (p. 705) widget to provide a combo widget.

Data Fields

- **Ewl_Widget * button**
- unsigned char **editable**:1
- **Ewl_Widget * header**
- **Ewl_MVC mvc**
- **Ewl_Widget * popup**
- unsigned char **scrollable**:1

7.23.2 Field Documentation

7.23.2.1 Ewl_Widget* Ewl_Combo::button

expand/contract button

7.23.2.2 unsigned char Ewl_Combo::editable

Is the combo editable

7.23.2.3 Ewl _Widget* Ewl _Combo::header

The combo header widget

7.23.2.4 Ewl _MVC Ewl _Combo::mvc

Inherit from **Ewl _MVC** (p. 705)

7.23.2.5 Ewl _Widget* Ewl _Combo::popup

Use a popup to display with.

7.23.2.6 unsigned char Ewl _Combo::scrollable

Is the combo scrollable

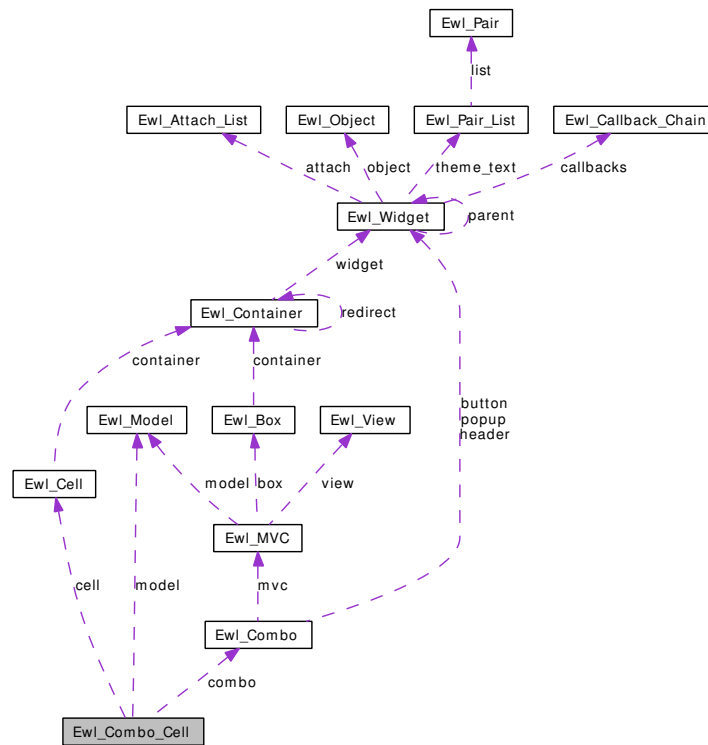
The documentation for this struct was generated from the following file:

- lib/ewl_combo.h

7.24 Ewl_Combo_Cell Struct Reference

```
#include <ewl_combo.h>
```

Collaboration diagram for Ewl_Combo_Cell:



7.24.1 Detailed Description

Inherits from `Ewl_Cell` (p. 596) and is used internally for the combo box.

Data Fields

- `Ewl_Cell` `cell`
- `Ewl_Combo *` `combo`
- `Ewl_Model *` `model`
- `void *` `mvc_data`

7.24.2 Field Documentation

7.24.2.1 `Ewl_Cell` `Ewl_Combo_Cell::cell`

Inherit from the cell

7.24.2.2 `Ewl_Combo *` `Ewl_Combo_Cell::combo`

The parent combo

7.24.2.3 Ewl_Model* Ewl_Combo_Cell::model

The model that was used to build the content of the cell

7.24.2.4 void* Ewl_Combo_Cell::mvc_data

The mvc data that was used to build the content of the cell

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

- lib/ewl_combo.h

7.25 Ewl_Config Struct Reference

```
#include <ewl_config.h>
```

7.25.1 Detailed Description

Contains Ewl configuration data.

Data Fields

- char * **app_name**
- struct {
 - Ecore_Hash * **instance**
 - Ecore_Hash * **system**
 - Ecore_Hash * **user**
- } **data**

7.25.2 Field Documentation

7.25.2.1 char* Ewl_Config::app_name

The application name

7.25.2.2 struct { ... } Ewl_Config::data

The config data

7.25.2.3 Ecore_Hash* Ewl_Config::instance

Data for this instance only

7.25.2.4 Ecore_Hash* Ewl_Config::system

System data

7.25.2.5 Ecore_Hash* Ewl_Config::user

User data

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

- lib/ewl_config.h

7.26 Ewl_Config_Cache Struct Reference

```
#include <ewl_debug.h>
```

7.26.1 Detailed Description

Contains cached Ewl configuration data.

Data Fields

- unsigned char **backtrace**:1
- unsigned char **enable**:1
- unsigned char **evas_render**:1
- unsigned char **gc_reap**:1
- int **level**
- unsigned char **print_keys**:1
- unsigned char **print_signals**:1
- unsigned char **segv**:1

7.26.2 Field Documentation

7.26.2.1 unsigned char Ewl_Config_Cache::backtrace

Print backtrace on D* messages

7.26.2.2 unsigned char Ewl_Config_Cache::enable

Enable debugging

7.26.2.3 unsigned char Ewl_Config_Cache::evas_render

Debug evas render calls

7.26.2.4 unsigned char Ewl_Config_Cache::gc_reap

Debug garbage collection

7.26.2.5 int Ewl_Config_Cache::level

Debug level

7.26.2.6 unsigned char Ewl_Config_Cache::print_keys

Print theem keys

7.26.2.7 unsigned char Ewl_Config_Cache::print_signals

Print theme signals

7.26.2.8 unsigned char Ewl_Config_Cache::segv

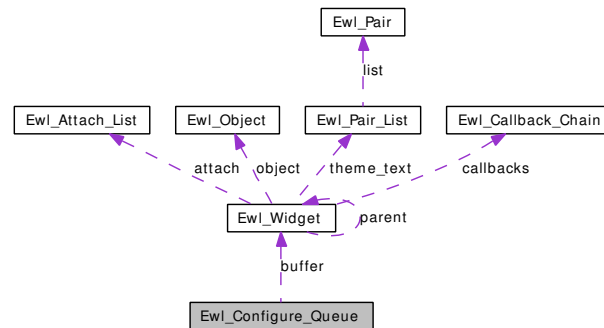
Segv on D* messages

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

- lib/ewl_debug.h

7.27 Ewl_Configure_Queue Struct Reference

Collaboration diagram for Ewl_Configure_Queue:



Data Fields

- `Ewl_Widget * buffer [EWL_CONFIGURE_QUEUE_SIZE]`
- `int end`

7.27.1 Field Documentation

7.27.1.1 `Ewl_Widget* Ewl_Configure_Queue::buffer[EWL_CONFIGURE_QUEUE_SIZE]`

7.27.1.2 `int Ewl_Configure_Queue::end`

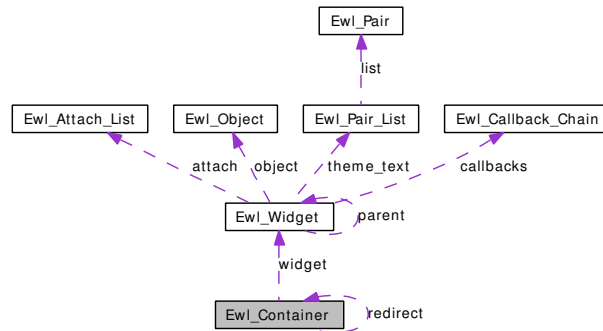
The documentation for this struct was generated from the following file:

- `lib/ewl_misc.c`

7.28 Ewl_Container Struct Reference

```
#include <ewl_container.h>
```

Collaboration diagram for Ewl_Container:



7.28.1 Detailed Description

Inherits from the **Ewl_Widget** (p. 785) and expands to allow for placing child widgets within the available space. Also adds notifiers for various child events.

Data Fields

- **Ewl_Child_Add** child_add
- **Ewl_Child_Hide** child_hide
- **Ewl_Child_Remove** child_remove
- **Ewl_Child_Resize** child_resize
- **Ewl_Child_Show** child_show
- **Ecore_DList *** children
- **void *** clip_box
- **Ewl_Container_Iterator** iterator
- **Ewl_Container *** redirect
- **Ewl_Widget** widget

7.28.2 Field Documentation

7.28.2.1 Ewl_Child_Add Ewl_Container::child_add

Function called on child add

7.28.2.2 Ewl_Child_Hide Ewl_Container::child_hide

Function called on child hide

7.28.2.3 Ewl_Child_Remove Ewl_Container::child_remove

Function called on child remove

7.28.2.4 Ewl_Child_Resize Ewl_Container::child_resize

Function called on child resize

7.28.2.5 Ewl_Child_Show Ewl_Container::child_show

Function called on child show

7.28.2.6 Ecore_DList* Ewl_Container::children

List of children that are contained.

7.28.2.7 void* Ewl_Container::clip_box

Clip box to bound widgets inside.

7.28.2.8 Ewl_Container_Iterator Ewl_Container::iterator

Function to find next child

7.28.2.9 Ewl_Container* Ewl_Container::redirect

Alternate parent for children

7.28.2.10 Ewl_Widget Ewl_Container::widget

Inherit the basics of the widget.

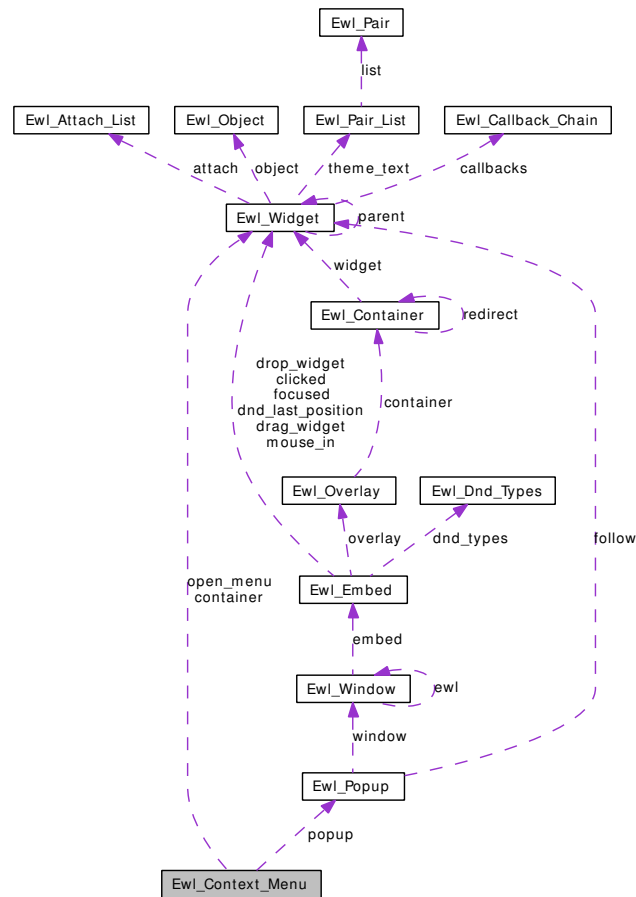
The documentation for this struct was generated from the following file:

- lib/ewl_container.h

7.29 Ewl_Context_Menu Struct Reference

```
#include <ewl_context_menu.h>
```

Collaboration diagram for Ewl_Context_Menu:



7.29.1 Detailed Description

Inherits from **Ewl_Popup** (p. 721) and extends to provide a context menu.

Data Fields

- **Ewl_Child_Add** child_add
- **Ewl_Child_Remove** child_remove
- **Ewl_Widget *** container
- **Ewl_Widget *** open_menu
- **Ewl_Popup** popup

7.29.2 Field Documentation

7.29.2.1 Ewl_Child_Add Ewl_Context_Menu::child_add

the overridden add cb of the container

7.29.2.2 Ewl_Child_Remove Ewl_Context_Menu::child_remove

7.29.2.3 Ewl_Widget* Ewl_Context_Menu::container

the container holding the children

7.29.2.4 Ewl_Widget* Ewl_Context_Menu::open_menu

a pointer to the current open submenu

7.29.2.5 Ewl_Popup Ewl_Context_Menu::popup

Inherit from **Ewl_Popup** (p. 721)

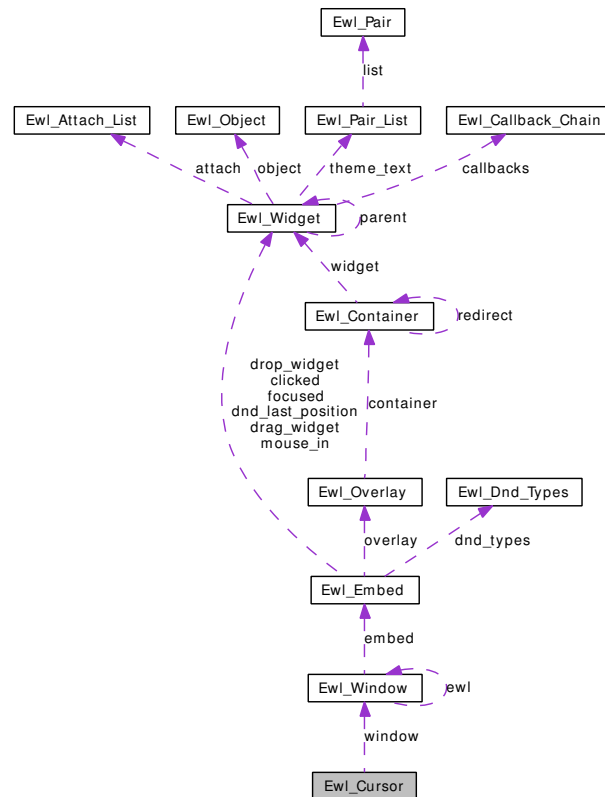
The documentation for this struct was generated from the following file:

- lib/ewl_context_menu.h

7.30 Ewl_Cursor Struct Reference

```
#include <ewl_cursor.h>
```

Collaboration diagram for Ewl_Cursor:



7.30.1 Detailed Description

Inherits from `Ewl_Window` (p. 788) to allow packing widgets as cursor objects.

Data Fields

- `int handle`
- `int refcount`
- `Ewl_Window window`

7.30.2 Field Documentation

7.30.2.1 `int Ewl_Cursor::handle`

Engine id for generated cursor

7.30.2.2 int Ewl_Cursor::refcount

Number of references to cursor

7.30.2.3 Ewl_Window Ewl_Cursor::window

Inherit from **Ewl_Window** (p. 788)

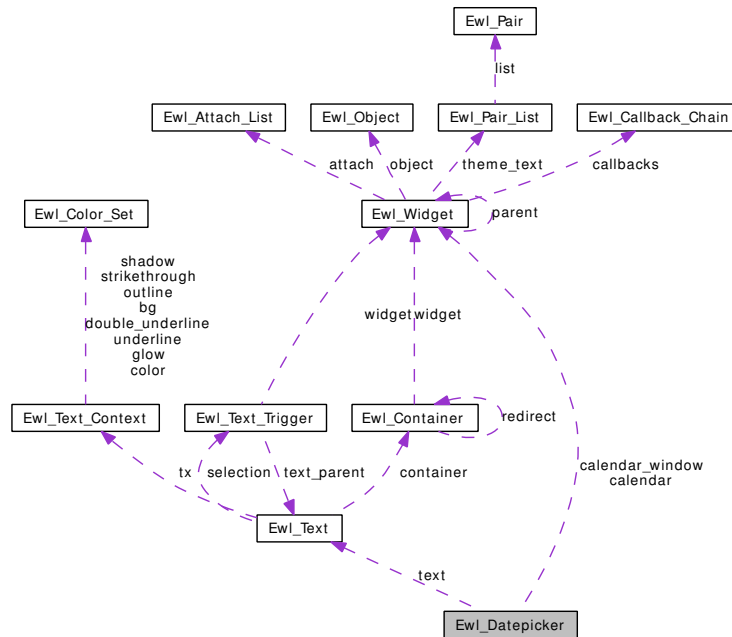
The documentation for this struct was generated from the following file:

- lib/ewl_cursor.h

7.31 Ewl_Datepicker Struct Reference

```
#include <ewl_datepicker.h>
```

Collaboration diagram for Ewl_Datepicker:



7.31.1 Detailed Description

Inherits from `Ewl_Text` (p. 757) and extends to provide a date picker.

Data Fields

- `Ewl_Widget * calendar`
- `Ewl_Widget * calendar_window`
- `Ewl_Text text`

7.31.2 Field Documentation

7.31.2.1 `Ewl_Widget* Ewl_Datepicker::calendar`

The calendar

7.31.2.2 `Ewl_Widget* Ewl_Datepicker::calendar_window`

The window

7.31.2.3 Ewl_Text Ewl_Datepicker::text

Inherit from **Ewl_Text** (p. 757)

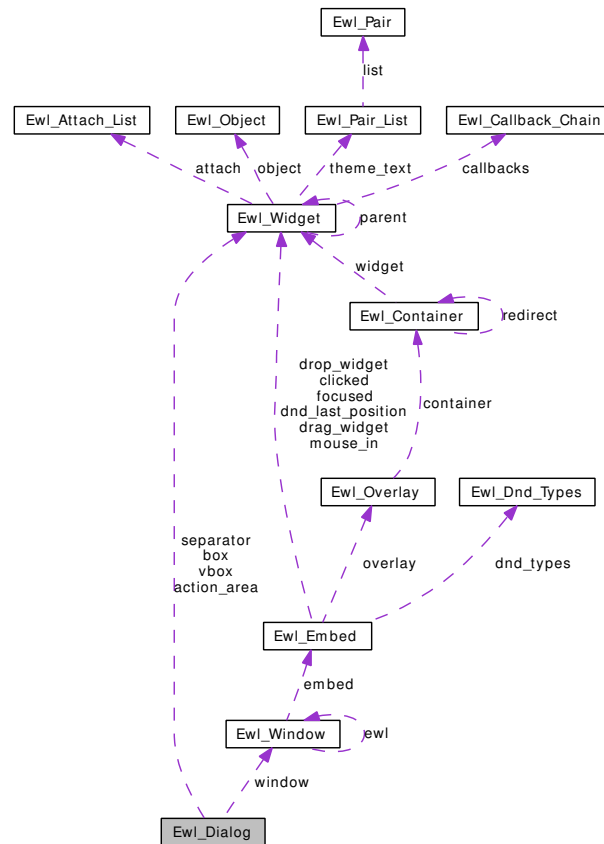
The documentation for this struct was generated from the following file:

- lib/ewl_datepicker.h

7.32 Ewl_Dialog Struct Reference

```
#include <ewl_dialog.h>
```

Collaboration diagram for Ewl_Dialog:



7.32.1 Detailed Description

Inherits from **Ewl_Window** (p. 788) and extends to provide a dialog widget.

Data Fields

- **Ewl_Widget** * `action_area`
- **Ewl_Position** `active_area`
- **Ewl_Widget** * `box`
- **Ewl_Position** `position`
- **Ewl_Widget** * `separator`
- **Ewl_Widget** * `vbox`
- **Ewl_Window** `window`

7.32.2 Field Documentation

7.32.2.1 Ewl _Widget* Ewl _Dialog::action _area

The box where the buttons are added

7.32.2.2 Ewl _Position Ewl _Dialog::active _area

Which section of the dialog is active

7.32.2.3 Ewl _Widget* Ewl _Dialog::box

The box for window layout

7.32.2.4 Ewl _Position Ewl _Dialog::position

Position of the action _area

7.32.2.5 Ewl _Widget* Ewl _Dialog::separator

The separator between vbox and action _area

7.32.2.6 Ewl _Widget* Ewl _Dialog::vbox

The box where messages are displayed

7.32.2.7 Ewl _Window Ewl _Dialog::window

Inherit from a window

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

- lib/ewl _dialog.h

7.33 Ewl_Dnd_Types Struct Reference

```
#include <ewl_events.h>
```

7.33.1 Detailed Description

Provides type information about an external DND drag.

Data Fields

- int **num_types**
- char ** **types**

7.33.2 Field Documentation

7.33.2.1 int Ewl_Dnd_Types::num_types

The number of DND types

7.33.2.2 char** Ewl_Dnd_Types::types

The list of DND types

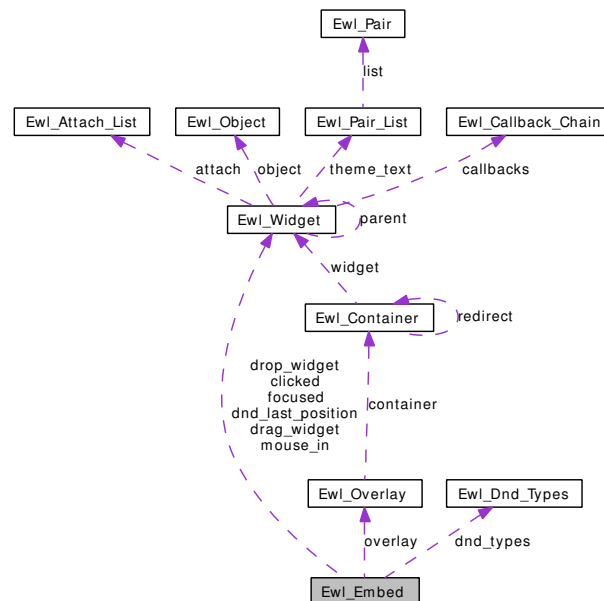
The documentation for this struct was generated from the following file:

- lib/**ewl_events.h**

7.34 Ewl_Embed Struct Reference

```
#include <ewl_embed.h>
```

Collaboration diagram for Ewl_Embed:



7.34.1 Detailed Description

Inherits from **Ewl_Overlay** (p. 712) that acts as a top level widget for interacting with the canvas.

Data Fields

- void * **canvas**
- void * **canvas_window**
- int **cursor**
- int **dnd_count**
- Ewl_Widget * **dnd_last_position**
- Ewl_Dnd_Types **dnd_types**
- void * **engine**
- const char * **engine_name**
- void * **ev_clip**
- int **focus**
- struct {
 - Ewl_Widget * **clicked**
 - Ewl_Widget * **drag_widget**
 - Ewl_Widget * **drop_widget**
 - Ewl_Widget * **focused**
 - Ewl_Widget * **mouse_in**
 } **last**

- `Ecore_Hash * obj_cache`
- `Ewl_Overlay overlay`
- `void * smart`
- `Ecore_DList * tab_order`
- `int x`
- `int y`

7.34.2 Field Documentation

7.34.2.1 `void* Ewl_Embed::canvas`

Canvas where drawing takes place.

7.34.2.2 `void* Ewl_Embed::canvas_window`

The window holding the canvas.

7.34.2.3 `Ewl_Widget* Ewl_Embed::clicked`

Last clicked widget

7.34.2.4 `int Ewl_Embed::cursor`

Id for current cursor in use

7.34.2.5 `int Ewl_Embed::dnd_count`

DND aware widget count

7.34.2.6 `Ewl_Widget* Ewl_Embed::dnd_last_position`

The last dnd position

7.34.2.7 `Ewl_Dnd_Types Ewl_Embed::dnd_types`

The dnd type

7.34.2.8 `Ewl_Widget* Ewl_Embed::drag_widget`

The current DND drag source

7.34.2.9 `Ewl_Widget* Ewl_Embed::drop_widget`

The current DND drop target

7.34.2.10 void* Ewl__Embed::engine

The display engine

7.34.2.11 const char* Ewl__Embed::engine__name

Name of current engine used

7.34.2.12 void* Ewl__Embed::ev__clip

Clip box to receive canvas events

7.34.2.13 int Ewl__Embed::focus

Indicates if it receives focus

7.34.2.14 Ewl__Widget* Ewl__Embed::focused

Last focused widget

7.34.2.15 struct { ... } Ewl__Embed::last

Collection of widgets to last receive events

7.34.2.16 Ewl__Widget* Ewl__Embed::mouse__in

Last widget to receive a mouse__in

7.34.2.17 Ecore__Hash* Ewl__Embed::obj__cache

Hash of object queues for reuse

7.34.2.18 Ewl__Overlay Ewl__Embed::overlay

Inherits from the **Ewl__Overlay** (p. 712) class

7.34.2.19 void* Ewl__Embed::smart

Manipulate **Ewl__Embed** (p. 629) from canvas

7.34.2.20 Ecore__DList* Ewl__Embed::tab__order

Order of widgets to send focus

7.34.2.21 int Ewl__Embed::x

Screen relative horizontal position of window

7.34.2.22 `int Ewl_Embed::y`

Screen relative vertical position of window

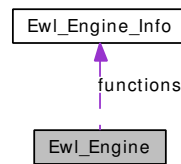
The documentation for this struct was generated from the following file:

- `lib/ewl_embed.h`

7.35 Ewl_Engine Struct Reference

```
#include <ewl_engines.h>
```

Collaboration diagram for Ewl_Engine:



7.35.1 Detailed Description

Contains information on an Ewl engine.

Data Fields

- `Ecore_DList * dependencies`
- `Ewl_Engine_Info * functions`
- `void * handle`
- `char * name`

7.35.2 Field Documentation

7.35.2.1 `Ecore_DList* Ewl_Engine::dependencies`

The engines this one depends on

7.35.2.2 `Ewl_Engine_Info* Ewl_Engine::functions`

The functions used by ewl, supplied by the engine

7.35.2.3 `void* Ewl_Engine::handle`

The dlsym handle

7.35.2.4 `char* Ewl_Engine::name`

The name of the engine

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

- `lib/ewl_engines.h`

7.36 Ewl_Engine_Info Struct Reference

```
#include <ewl_engines.h>
```

7.36.1 Detailed Description

Contains the function pointers provided by an engine.

Data Fields

- struct {
 void ** **canvas**
 void ** **pointer**
 void ** **theme**
 void ** **window**
} **hooks**
- void(* **shutdown**)(Ewl_Engine *engine)

7.36.2 Field Documentation

7.36.2.1 void** Ewl_Engine_Info::canvas

The canvas hooks

7.36.2.2 struct { ... } Ewl_Engine_Info::hooks

The engine hooks

7.36.2.3 void** Ewl_Engine_Info::pointer

The pointer hooks

7.36.2.4 void(* Ewl_Engine_Info::shutdown)(Ewl_Engine *engine)

Shutdown the engine

7.36.2.5 void** Ewl_Engine_Info::theme

The theme hooks

7.36.2.6 void** Ewl_Engine_Info::window

The window hooks

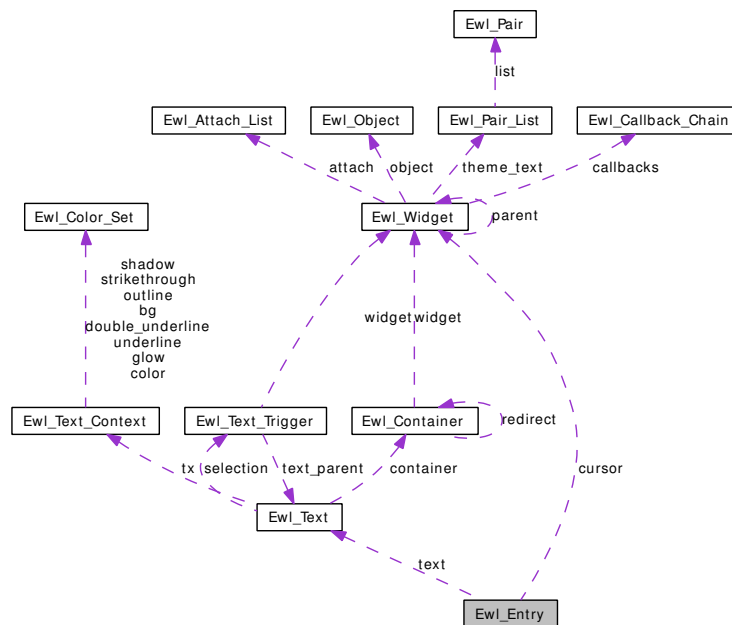
The documentation for this struct was generated from the following file:

- lib/ewl_engines.h

7.37 Ewl_Entry Struct Reference

```
#include <ewl_entry.h>
```

Collaboration diagram for Ewl_Entry:



7.37.1 Detailed Description

Inherits from `Ewl_Text` (p. 757) and extends to allow editing of the text.

Data Fields

- `Ewl_Widget * cursor`
- unsigned int `editable`
- unsigned int `in_select_mode`
- unsigned int `multiline`
- `Ewl_Text text`

7.37.2 Field Documentation

7.37.2.1 `Ewl_Widget* Ewl_Entry::cursor`

The cursor

7.37.2.2 unsigned int `Ewl_Entry::editable`

Is the entry editable

7.37.2.3 unsigned int Ewl_Entry::in_select_mode

Are we in select mode

7.37.2.4 unsigned int Ewl_Entry::multiline

Is the entry multi-line

7.37.2.5 Ewl_Text Ewl_Entry::text

The **Ewl_Text** (p. 757) parent

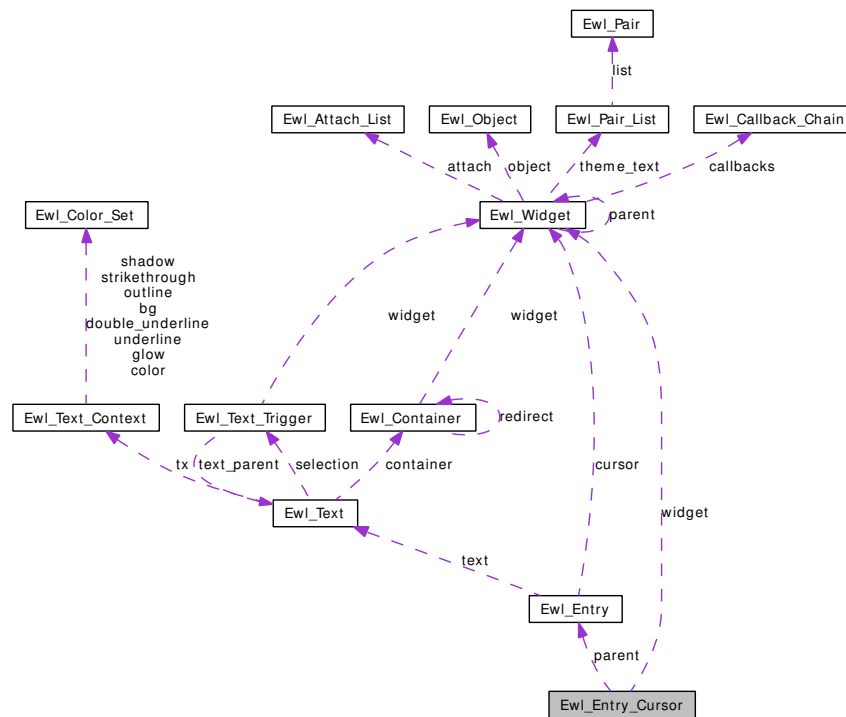
The documentation for this struct was generated from the following file:

- lib/ewl_entry.h

7.38 Ewl_Entry_Cursor Struct Reference

```
#include <ewl_entry.h>
```

Collaboration diagram for Ewl_Entry_Cursor:



7.38.1 Detailed Description

Inherit from **Ewl_Widget** (p. 785) and extend for the Entry cursor.

Data Fields

- **Ewl_Entry** * parent
- **Ewl_Widget** widget

7.38.2 Field Documentation

7.38.2.1 Ewl_Entry* Ewl_Entry_Cursor::parent

The parent **Ewl_Text** (p. 757)

7.38.2.2 Ewl_Widget Ewl_Entry_Cursor::widget

Inherit from **Ewl_Widget** (p. 785)

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

- `lib/ewl_entry.h`

7.39 Ewl_Event_Action_Response Struct Reference

```
#include <ewl_events.h>
```

7.39.1 Detailed Description

Stores the reponse from a user action.

Data Fields

- unsigned int **response**

7.39.2 Field Documentation

7.39.2.1 unsigned int Ewl_Event_Action_Response::response

The response ID

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

- lib/**ewl_events.h**

7.40 Ewl_Event_Dnd_Data_Received Struct Reference

```
#include <ewl_events.h>
```

7.40.1 Detailed Description

Provides information about dnd drop data.

Data Fields

- void * **data**
- unsigned int **format**
- unsigned int **len**
- char * **type**

7.40.2 Field Documentation

7.40.2.1 void* Ewl_Event_Dnd_Data_Received::data

Data from drop source

7.40.2.2 unsigned int Ewl_Event_Dnd_Data_Received::format

Bit format of received data

7.40.2.3 unsigned int Ewl_Event_Dnd_Data_Received::len

Length of received data

7.40.2.4 char* Ewl_Event_Dnd_Data_Received::type

Type of data from drop source

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

- lib/**ewl_events.h**

7.41 Ewl_Event_Dnd_Data_Request Struct Reference

```
#include <ewl_events.h>
```

7.41.1 Detailed Description

Provides information about dnd data requests.

Data Fields

- void * **handle**
- char * **type**

7.41.2 Field Documentation

7.41.2.1 void* Ewl_Event_Dnd_Data_Request::handle

Engine specific handle for responding to request

7.41.2.2 char* Ewl_Event_Dnd_Data_Request::type

Type of data requested

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

- lib/**ewl_events.h**

7.42 Ewl_Event_Dnd_Drop Struct Reference

```
#include <ewl_events.h>
```

7.42.1 Detailed Description

Provides information about dnd drops movement.

Data Fields

- void * **data**
- int **x**
- int **y**

7.42.2 Field Documentation

7.42.2.1 void* Ewl_Event_Dnd_Drop::data

Data from drop source

7.42.2.2 int Ewl_Event_Dnd_Drop::x

X coordinate the mouse moved to

7.42.2.3 int Ewl_Event_Dnd_Drop::y

Y coordinate the mouse moved to

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

- lib/ewl_events.h

7.43 Ewl_Event_Dnd_Position Struct Reference

```
#include <ewl_events.h>
```

7.43.1 Detailed Description

Provides information about dnd drops movement.

Data Fields

- int x
- int y

7.43.2 Field Documentation

7.43.2.1 int Ewl_Event_Dnd_Position::x

X coordinate the mouse moved to

7.43.2.2 int Ewl_Event_Dnd_Position::y

Y coordinate the mouse moved to

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

- lib/ewl_events.h

7.44 Ewl_Event_Key Struct Reference

```
#include <ewl_events.h>
```

7.44.1 Detailed Description

Provides the basic information about key events.

Data Fields

- char * **keyname**
- unsigned int **modifiers**

7.44.2 Field Documentation

7.44.2.1 char* Ewl_Event_Key::keyname

Name of the key that was pressed

7.44.2.2 unsigned int Ewl_Event_Key::modifiers

Key modifiers that were pressed

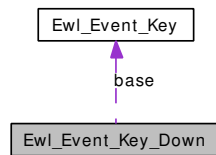
The documentation for this struct was generated from the following file:

- lib/**ewl_events.h**

7.45 Ewl_Event_Key_Down Struct Reference

```
#include <ewl_events.h>
```

Collaboration diagram for Ewl_Event_Key_Down:



7.45.1 Detailed Description

Provides clients with necessary information about the key press event.

Data Fields

- Ewl_Event_Key base

7.45.2 Field Documentation

7.45.2.1 Ewl_Event_Key Ewl_Event_Key_Down::base

Basic key information

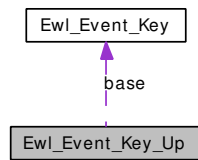
The documentation for this struct was generated from the following file:

- lib/ewl_events.h

7.46 Ewl_Event_Key_Up Struct Reference

```
#include <ewl_events.h>
```

Collaboration diagram for Ewl_Event_Key_Up:



7.46.1 Detailed Description

Provides clients with necessary information about the key release event.

Data Fields

- Ewl_Event_Key base

7.46.2 Field Documentation

7.46.2.1 Ewl_Event_Key Ewl_Event_Key_Up::base

Basic key information

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

- lib/ewl_events.h

7.47 Ewl_Event_Mouse Struct Reference

```
#include <ewl_events.h>
```

7.47.1 Detailed Description

Provides base mouse information.

Data Fields

- unsigned int **modifiers**
- int **x**
- int **y**

7.47.2 Field Documentation

7.47.2.1 unsigned int Ewl_Event_Mouse::modifiers

Modifiers that were pressed

7.47.2.2 int Ewl_Event_Mouse::x

X coordinate the mouse release occurred at

7.47.2.3 int Ewl_Event_Mouse::y

Y coordinate the mouse release occurred at

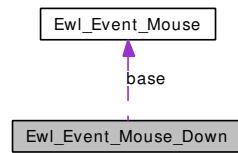
The documentation for this struct was generated from the following file:

- lib/**ewl_events.h**

7.48 Ewl_Event_Mouse_Down Struct Reference

```
#include <ewl_events.h>
```

Collaboration diagram for Ewl_Event_Mouse_Down:



7.48.1 Detailed Description

Provides information about the mouse down event.

Data Fields

- `Ewl_Event_Mouse` `base`
- `int` `button`
- `int` `clicks`

7.48.2 Field Documentation

7.48.2.1 `Ewl_Event_Mouse` `Ewl_Event_Mouse_Down::base`

Base mouse information

7.48.2.2 `int` `Ewl_Event_Mouse_Down::button`

The mouse button that was released

7.48.2.3 `int` `Ewl_Event_Mouse_Down::clicks`

Number of consecutive clicks

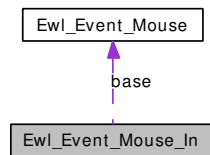
The documentation for this struct was generated from the following file:

- `lib/ewl_events.h`

7.49 Ewl_Event_Mouse_In Struct Reference

```
#include <ewl_events.h>
```

Collaboration diagram for Ewl_Event_Mouse_In:



7.49.1 Detailed Description

Provides information about the mouse entering.

Data Fields

- Ewl_Event_Mouse base

7.49.2 Field Documentation

7.49.2.1 Ewl_Event_Mouse Ewl_Event_Mouse_In::base

Base mouse information

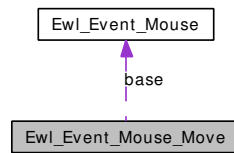
The documentation for this struct was generated from the following file:

- lib/ewl_events.h

7.50 Ewl_Event_Mouse_Move Struct Reference

```
#include <ewl_events.h>
```

Collaboration diagram for Ewl_Event_Mouse_Move:



7.50.1 Detailed Description

Provides information about mouse movement.

Data Fields

- **Ewl_Event_Mouse base**

7.50.2 Field Documentation

7.50.2.1 Ewl_Event_Mouse Ewl_Event_Mouse_Move::base

Base mouse information

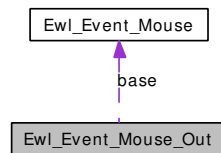
The documentation for this struct was generated from the following file:

- **lib/ewl_events.h**

7.51 Ewl_Event_Mouse_Out Struct Reference

```
#include <ewl_events.h>
```

Collaboration diagram for Ewl_Event_Mouse_Out:



7.51.1 Detailed Description

Provides information about the mouse leaving.

Data Fields

- Ewl_Event_Mouse base

7.51.2 Field Documentation

7.51.2.1 Ewl_Event_Mouse Ewl_Event_Mouse_Out::base

Base mouse information

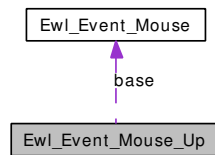
The documentation for this struct was generated from the following file:

- lib/ewl_events.h

7.52 Ewl_Event_Mouse_Up Struct Reference

```
#include <ewl_events.h>
```

Collaboration diagram for Ewl_Event_Mouse_Up:



7.52.1 Detailed Description

Provides information about the mouse up event.

Data Fields

- **Ewl_Event_Mouse** base
- **int** button

7.52.2 Field Documentation

7.52.2.1 Ewl_Event_Mouse Ewl_Event_Mouse_Up::base

Base mouse information

7.52.2.2 int Ewl_Event_Mouse_Up::button

The mouse button that was released

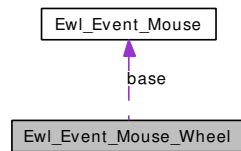
The documentation for this struct was generated from the following file:

- **lib/ewl_events.h**

7.53 Ewl_Event_Mouse_Wheel Struct Reference

```
#include <ewl_events.h>
```

Collaboration diagram for Ewl_Event_Mouse_Wheel:



7.53.1 Detailed Description

Provides information about the mouse wheel scrolling.

Data Fields

- Ewl_Event_Mouse base
- int dir
- int z

7.53.2 Field Documentation

7.53.2.1 Ewl_Event_Mouse Ewl_Event_Mouse_Wheel::base

Base mouse information

7.53.2.2 int Ewl_Event_Mouse_Wheel::dir

Direction mouse wheel scrolled

7.53.2.3 int Ewl_Event_Mouse_Wheel::z

Z value of mouse wheel

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

- lib/ewl_events.h

7.54 Ewl_Event_Window_Expose Struct Reference

```
#include <ewl_events.h>
```

7.54.1 Detailed Description

Describes the canvas region that received an expose notification.

Data Fields

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

7.54.2 Field Documentation

7.54.2.1 int Ewl_Event_Window_Expose::h

Height of the expose rectangle

7.54.2.2 int Ewl_Event_Window_Expose::w

Width of the expose rectangle

7.54.2.3 int Ewl_Event_Window_Expose::x

Beginning X coordinate for the expose rectangle

7.54.2.4 int Ewl_Event_Window_Expose::y

Beginning Y coordinate for the expose rectangle

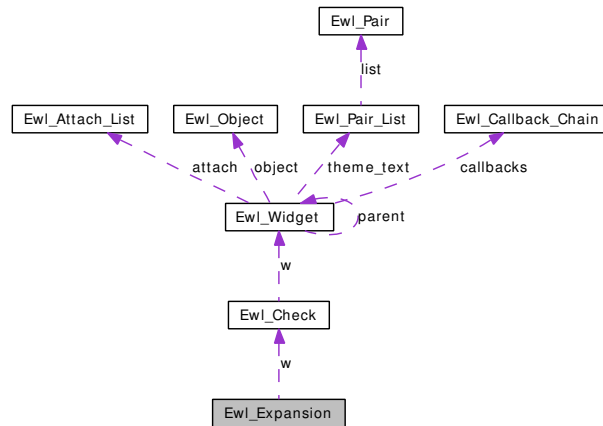
The documentation for this struct was generated from the following file:

- lib/**ewl_events.h**

7.55 Ewl_Expansion Struct Reference

```
#include <ewl_expansion.h>
```

Collaboration diagram for Ewl_Expansion:



7.55.1 Detailed Description

Inherits from **Ewl_Check** (p. 597) and extends it to provide a expanded state.

Data Fields

- **int** **expandable**
- **Ewl_Check** **w**

7.55.2 Field Documentation

7.55.2.1 **int** **Ewl_Expansion::expandable**

Indicates if this node is expandable or not

7.55.2.2 **Ewl_Check** **Ewl_Expansion::w**

Inherit from **Ewl_Check** (p. 597)

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

- **lib/ewl_expansion.h**

7.56.2 Field Documentation

7.56.2.1 Ewl_Dialog Ewl_Filedialog::dialog

The dialog base class

7.56.2.2 Ewl_Widget* Ewl_Filedialog::fp

Ewl_Filepicker (p. 668)

7.56.2.3 Ewl_Widget* Ewl_Filedialog::menu

The menu

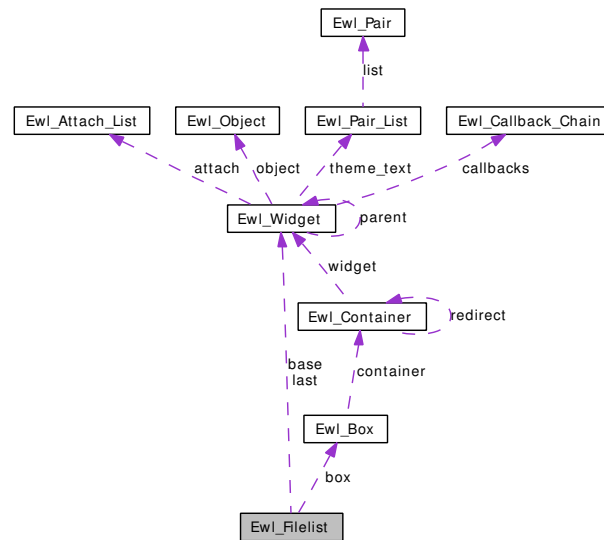
The documentation for this struct was generated from the following file:

- lib/ewl_filedialog.h

7.57 Ewl_Filelist Struct Reference

```
#include <ewl_filelist.h>
```

Collaboration diagram for Ewl_Filelist:



7.57.1 Detailed Description

Inherits from **Ewl_Box** (p.586) and extends to display a file list.

Data Fields

- **Ewl_Box** box
- void(* **dir_change**)(Ewl_Filelist *fl)
- char * **directory**
- const char *(* **file_name_get**)(Ewl_Filelist *fl, void *file)
- char * **filter**
- void(* **filter_change**)(Ewl_Filelist *fl)
- unsigned char **multiselect**:1
- void(* **multiselect_change**)(Ewl_Filelist *fl)
- struct {
 - Ewl_Scrollpane_Flags h
 - Ewl_Scrollpane_Flags v
 } **scroll_flags**
- struct {
 - Ewl_Widget * base
 - Ewl_Widget * last
 } **select**
- Ecore_List * **selected**
- void(* **selected_file_add**)(Ewl_Filelist *fl, const char *file)

- void(* selected_unselect)(Ewl_Filelist *fl)
- void(* shift_handle)(Ewl_Filelist *fl, Ewl_Widget *clicked)
- void(* show_dot_change)(Ewl_Filelist *fl)
- unsigned char show_dot_files:1

7.57.2 Field Documentation

7.57.2.1 Ewl_Widget* Ewl_Filelist::base

First select in SHIFT select

7.57.2.2 Ewl_Box Ewl_Filelist::box

Inherits from **Ewl_Box** (p. 586)

7.57.2.3 void(* Ewl_Filelist::dir_change)(Ewl_Filelist *fl)

Callback to notify of directory change

7.57.2.4 char* Ewl_Filelist::directory

The directory to display

7.57.2.5 const char*(* Ewl_Filelist::file_name_get)(Ewl_Filelist *fl, void *file)

Callback to get the selected filename

7.57.2.6 char* Ewl_Filelist::filter

The file filter to employ

7.57.2.7 void(* Ewl_Filelist::filter_change)(Ewl_Filelist *fl)

Callback to notify of filter change

7.57.2.8 Ewl_Scrollpane_Flags Ewl_Filelist::h

Horizontal scroll flag

7.57.2.9 Ewl_Widget* Ewl_Filelist::last

Last selected in SHIFT select

7.57.2.10 unsigned char Ewl_Filelist::multiselect

Allow multiple file selctions

7.57.2.11 void(* Ewl_Filelist::multiselect_change)(Ewl_Filelist *fl)

Callback to notify of multiselect state change

7.57.2.12 struct { ... } Ewl_Filelist::scroll_flags

Flags to modify a containing scrollpane

7.57.2.13 struct { ... } Ewl_Filelist::select

Data used in SHIFT select

7.57.2.14 Ecore_List* Ewl_Filelist::selected

The selected files

7.57.2.15 void(* Ewl_Filelist::selected_file_add)(Ewl_Filelist *fl, const char *file)

Callback to notify of a change to the selected files

7.57.2.16 void(* Ewl_Filelist::selected_unselect)(Ewl_Filelist *fl)

Callback to unselect all files

7.57.2.17 void(* Ewl_Filelist::shift_handle)(Ewl_Filelist *fl, Ewl_Widget *clicked)

Callback to handle SHIFT clicks

7.57.2.18 void(* Ewl_Filelist::show_dot_change)(Ewl_Filelist *fl)

Callback to notify of show dot file setting change

7.57.2.19 unsigned char Ewl_Filelist::show_dot_files

Show . files

7.57.2.20 Ewl_Scrollpane_Flags Ewl_Filelist::v

Vertical scroll flag

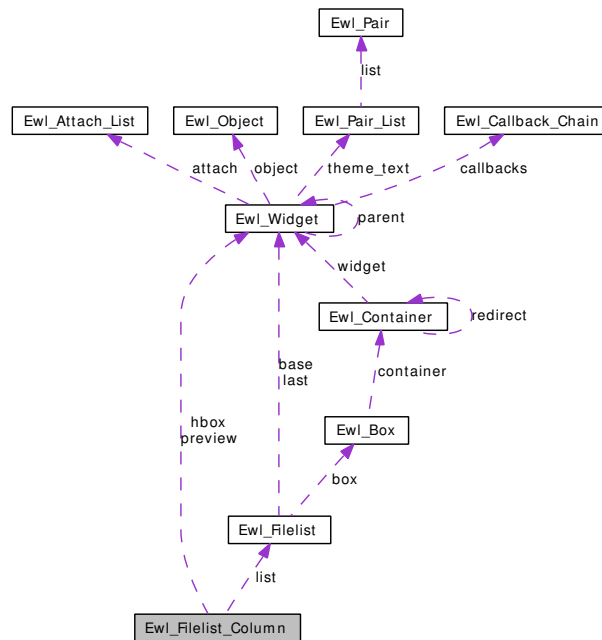
The documentation for this struct was generated from the following file:

- lib/ewl_filelist.h

7.58 Ewl_Filelist_Column Struct Reference

```
#include <ewl_filelist_column.h>
```

Collaboration diagram for Ewl_Filelist_Column:



7.58.1 Detailed Description

Inherits from **Ewl_Filelist** (p. 658) and extends to provide a column view.

Data Fields

- **Ecore_List * dirs**
- **Ewl_Widget * hbox**
- **Ewl_Filelist list**
- **Ewl_Widget * preview**

7.58.2 Field Documentation

7.58.2.1 Ecore_List* Ewl_Filelist_Column::dirs

List of directories shown

7.58.2.2 Ewl_Widget* Ewl_Filelist_Column::hbox

Hbox to contain the file lists

7.58.2.3 Ewl_Filelist Ewl_Filelist_Column::list

Inherit from filelist

7.58.2.4 Ewl_Widget* Ewl_Filelist_Column::preview

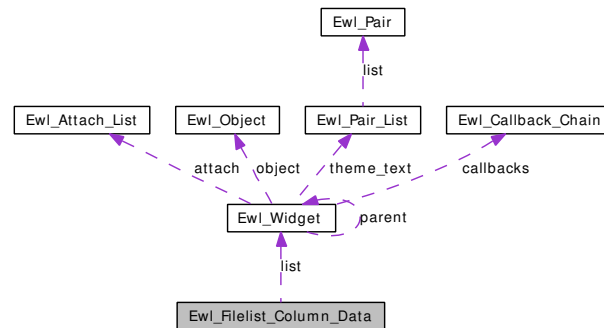
The preview

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

- lib/ewl_filelist_column.h

7.59 Ewl_Filelist_Column_Data Struct Reference

Collaboration diagram for Ewl_Filelist_Column_Data:



7.59.1 Detailed Description

Stores information on the filelist column data.

Data Fields

- `char * dir`
- `Ewl_Widget * list`

7.59.2 Field Documentation

7.59.2.1 `char* Ewl_Filelist_Column_Data::dir`

7.59.2.2 `Ewl_Widget* Ewl_Filelist_Column_Data::list`

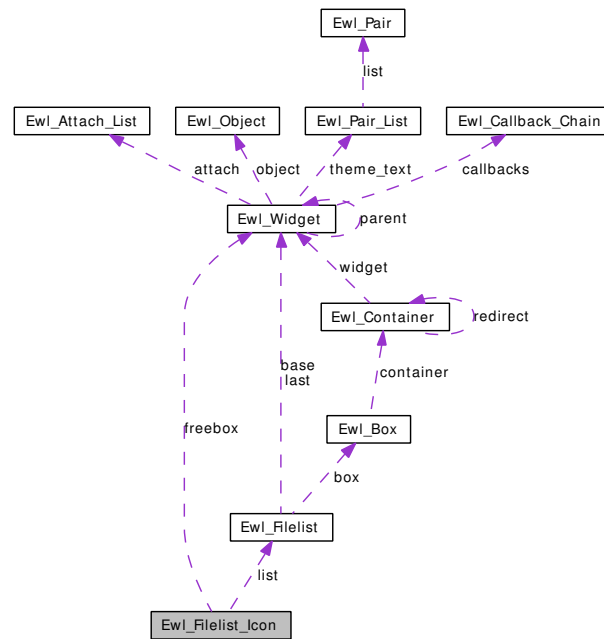
The documentation for this struct was generated from the following file:

- `lib/ewl_filelist_column.c`

7.60 Ewl_Filelist_Icon Struct Reference

```
#include <ewl_filelist_icon.h>
```

Collaboration diagram for Ewl_Filelist_Icon:



7.60.1 Detailed Description

Inherits from **Ewl_Filelist** (p. 658) and extends for an icon view.

Data Fields

- **Ewl_Widget * freebox**
- **Ewl_Filelist list**

7.60.2 Field Documentation

7.60.2.1 Ewl_Widget* Ewl_Filelist_Icon::freebox

Freebox to hold icons

7.60.2.2 Ewl_Filelist Ewl_Filelist_Icon::list

Inherit from filelist

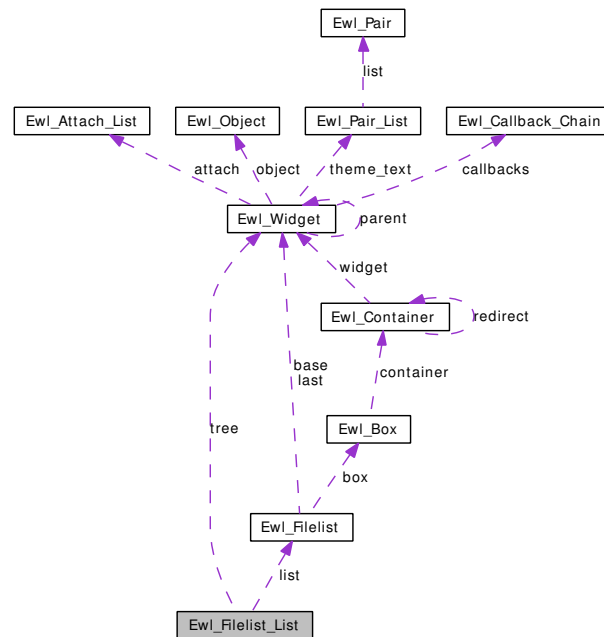
The documentation for this struct was generated from the following file:

- **lib/ewl_filelist_icon.h**

7.61 Ewl_Filelist_List Struct Reference

```
#include <ewl_filelist_list.h>
```

Collaboration diagram for Ewl_Filelist_List:



7.61.1 Detailed Description

Inherits from **Ewl_Filelist** (p. 658) and extends to create a list view.

Data Fields

- **Ewl_Filelist list**
- **Ewl_Widget * tree**

7.61.2 Field Documentation

7.61.2.1 Ewl_Filelist Ewl_Filelist_List::list

Inherit from filelist

7.61.2.2 Ewl_Widget* Ewl_Filelist_List::tree

Tree to hold icons

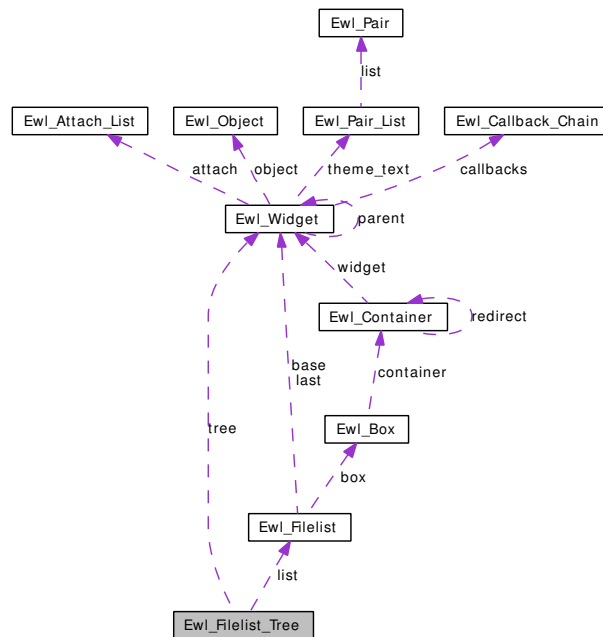
The documentation for this struct was generated from the following file:

- **lib/ewl_filelist_list.h**

7.62 Ewl_Filelist_Tree Struct Reference

```
#include <ewl_filelist_tree.h>
```

Collaboration diagram for Ewl_Filelist_Tree:



7.62.1 Detailed Description

Inherits from **Ewl_Filelist** (p. 658) and extends to provide a tree view.

Data Fields

- **Ewl_Filelist** list
- **Ewl_Widget *** tree

7.62.2 Field Documentation

7.62.2.1 Ewl_Filelist Ewl_Filelist_Tree::list

Inherit from filelist

7.62.2.2 Ewl_Widget* Ewl_Filelist_Tree::tree

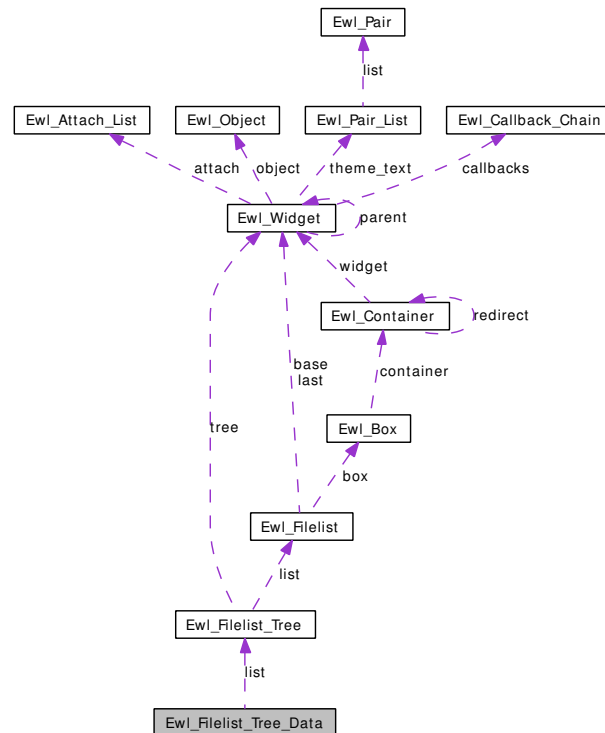
Tree to hold icons

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

- **lib/ewl_filelist_tree.h**

7.63 Ewl_Filelist_Tree_Data Struct Reference

Collaboration diagram for Ewl_Filelist_Tree_Data:



7.63.1 Detailed Description

Contains information on a filelist tree data.

Data Fields

- `Ecore_List * files`
- `Ewl_Filelist_Tree * list`

7.63.2 Field Documentation

7.63.2.1 `Ecore_List* Ewl_Filelist_Tree_Data::files`

7.63.2.2 `Ewl_Filelist_Tree* Ewl_Filelist_Tree_Data::list`

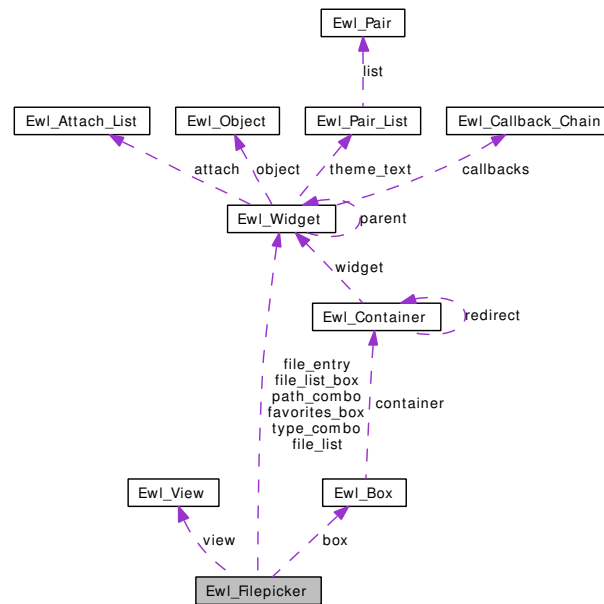
The documentation for this struct was generated from the following file:

- `lib/ewl_filelist_tree.c`

7.64 Ewl_Filepicker Struct Reference

```
#include <ewl_filepicker.h>
```

Collaboration diagram for Ewl_Filepicker:



7.64.1 Detailed Description

Inherits from **Ewl_Box** (p. 586) and extends to provide a filepicker.

Data Fields

- **Ewl_Box** box
- **Ewl_Widget** * favorites_box
- **Ewl_Widget** * file_entry
- **Ewl_Widget** * file_list
- **Ewl_Widget** * file_list_box
- **Ecore_List** * filters
- **Ecore_List** * path
- **Ewl_Widget** * path_combo
- unsigned char show_favorites:1
- **Ewl_Widget** * type_combo
- **Ewl_View** * view

7.64.2 Field Documentation

7.64.2.1 Ewl_Box Ewl_Filepicker::box

Inherit from **Ewl_Box** (p. 586)

7.64.2.2 Ewl_Widget* Ewl_Filepicker::favorites_box

Favoirte directory box

7.64.2.3 Ewl_Widget* Ewl_Filepicker::file_entry

Filename entry box

7.64.2.4 Ewl_Widget* Ewl_Filepicker::file_list

The file list widet

7.64.2.5 Ewl_Widget* Ewl_Filepicker::file_list_box

Box to hold the file list widget

7.64.2.6 Ecore_List* Ewl_Filepicker::filters

The type filters

7.64.2.7 Ecore_List* Ewl_Filepicker::path

The path components

7.64.2.8 Ewl_Widget* Ewl_Filepicker::path_combo

Path combo box

7.64.2.9 unsigned char Ewl_Filepicker::show_favorites

Show the favorite box

7.64.2.10 Ewl_Widget* Ewl_Filepicker::type_combo

File type combo

7.64.2.11 Ewl_View* Ewl_Filepicker::view

The filelist view to use

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

- lib/ewl_filepicker.h

7.65 Ewl_Filepicker_Filter Struct Reference

7.65.1 Detailed Description

Contains information on a filepicker filter.

Data Fields

- char * **filter**
- char * **name**

7.65.2 Field Documentation

7.65.2.1 char* Ewl_Filepicker_Filter::filter

7.65.2.2 char* Ewl_Filepicker_Filter::name

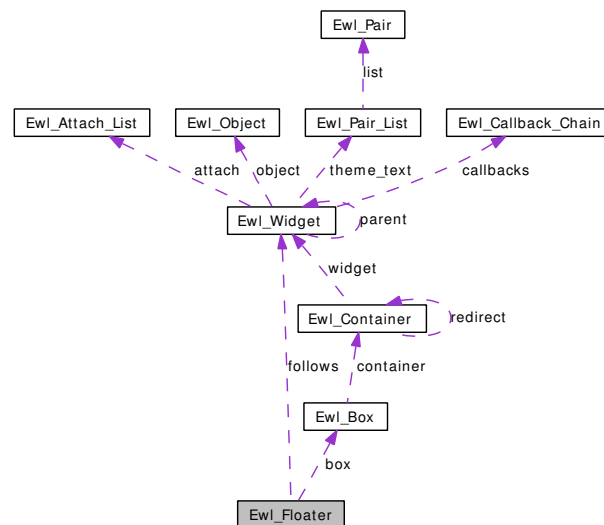
The documentation for this struct was generated from the following file:

- lib/**ewl_filepicker.c**

7.66 Ewl_Floater Struct Reference

```
#include <ewl_floater.h>
```

Collaboration diagram for Ewl_Floater:



7.66.1 Detailed Description

Inherits from **Ewl_Box** (p.586) for layout purposes. Provides a means to layer above other widgets and to follow the movement of another widget.

Data Fields

- **Ewl_Box** **box**
- **Ewl_Widget *** **follows**
- **int** **x**
- **int** **y**

7.66.2 Field Documentation

7.66.2.1 Ewl_Box Ewl_Floater::box

Inherit from **Ewl_Box** (p.586)

7.66.2.2 Ewl_Widget* Ewl_Floater::follows

The widget to position relative to

7.66.2.3 int Ewl_Floater::x

The x offset from follows x position

7.66.2.4 int Ewl_Floater::y

The y offset from follows y positon

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

- lib/ewl_floater.h

7.67.2.4 Ewl_Orientation Ewl_Freebox::orientation

The orientation of the box

7.67.2.5 unsigned short Ewl_Freebox::sorted

Is the box sorted

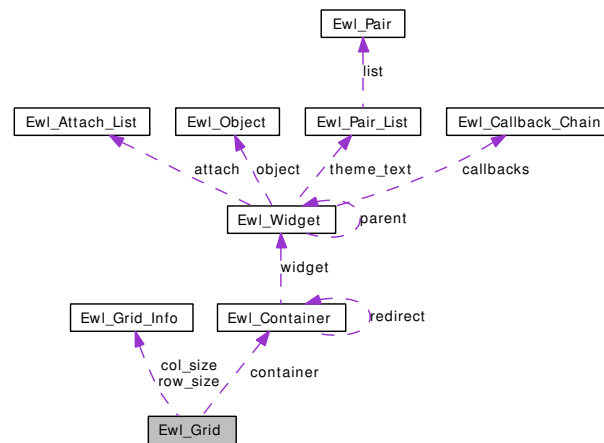
The documentation for this struct was generated from the following file:

- lib/**ewl_freebox.h**

7.68 Ewl_Grid Struct Reference

```
#include <ewl_grid.h>
```

Collaboration diagram for Ewl_Grid:



7.68.1 Detailed Description

Inherit from `Ewl_Container` (p. 618) and extend to provide a grid layout widget.

Data Fields

- `Ewl_Grid_Info * col_size`
- `int cols`
- `Ewl_Container container`
- `unsigned char data_dirty:1`
- `unsigned char homogeneous_h:1`
- `unsigned char homogeneous_v:1`
- `char * map`
- `Ewl_Orientation orientation`
- `Ewl_Grid_Info * row_size`
- `int rows`
- `int space`

7.68.2 Field Documentation

7.68.2.1 `Ewl_Grid_Info* Ewl_Grid::col_size`

Horizontal/vertical size of the columns

7.68.2.2 `int Ewl_Grid::cols`

Column count

7.68.2.3 Ewl_Container Ewl_Grid::container

Inherit from **Ewl_Container** (p. 618)

7.68.2.4 unsigned char Ewl_Grid::data_dirty

flag if the size of the columns and rows must be recalculated

7.68.2.5 unsigned char Ewl_Grid::homogeneous_h

Horizontal homogeneous flag

7.68.2.6 unsigned char Ewl_Grid::homogeneous_v

Vertical homogeneous flag

7.68.2.7 char* Ewl_Grid::map

Map of the child that have a postion

7.68.2.8 Ewl_Orientation Ewl_Grid::orientation

The grid orientation

7.68.2.9 Ewl_Grid_Info* Ewl_Grid::row_size

Horizontal/vertical size of the rows

7.68.2.10 int Ewl_Grid::rows

Row count

7.68.2.11 int Ewl_Grid::space

Space count

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

- lib/ewl_grid.h

7.69 Ewl_Grid_Child Struct Reference

```
#include <ewl_grid.h>
```

7.69.1 Detailed Description

Contains information about a grid child.

Data Fields

- int `end_col`
- int `end_row`
- int `start_col`
- int `start_row`

7.69.2 Field Documentation

7.69.2.1 int Ewl_Grid_Child::end_col

The end column

7.69.2.2 int Ewl_Grid_Child::end_row

The end row

7.69.2.3 int Ewl_Grid_Child::start_col

The start column

7.69.2.4 int Ewl_Grid_Child::start_row

The start row

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

- `lib/ewl_grid.h`

7.70 Ewl_Grid_Info Struct Reference

```
#include <ewl_grid.h>
```

7.70.1 Detailed Description

Contains information about a row or column of **Ewl_Grid** (p.675).

Data Fields

- int **current_pos**
- int **current_size**
- int **preferred_size**
- **Ewl_Grid_Resize_Type** **resize_type**
- union {
 - float **rel_size**
 - int **size**
- **user**

7.70.2 Field Documentation

7.70.2.1 int Ewl_Grid_Info::current_pos

the current position

7.70.2.2 int Ewl_Grid_Info::current_size

the current size

7.70.2.3 int Ewl_Grid_Info::preferred_size

The greatest preferred size of a widget inside

7.70.2.4 float Ewl_Grid_Info::rel_size

The relative size

7.70.2.5 Ewl_Grid_Resize_Type Ewl_Grid_Info::resize_type

Are there values set by the user

7.70.2.6 int Ewl_Grid_Info::size

The size set by the user

7.70.2.7 union { ... } Ewl_Grid_Info::user

The user set values for the grid

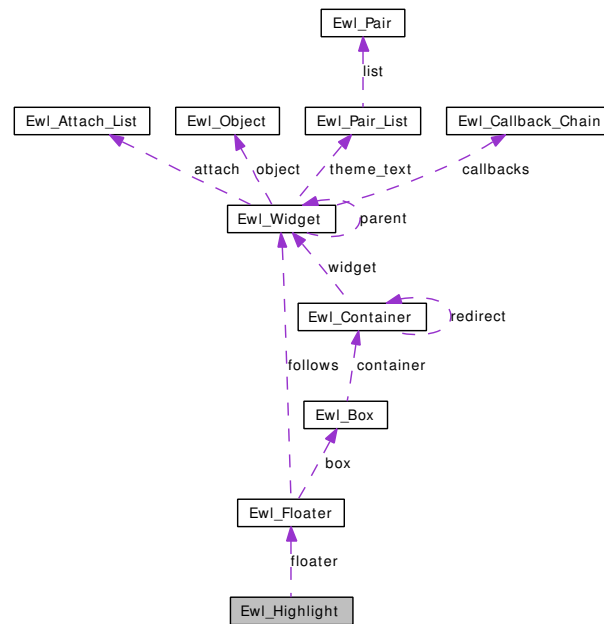
The documentation for this struct was generated from the following file:

- lib/ewl_grid.h

7.71 Ewl_Highlight Struct Reference

```
#include <ewl_highlight.h>
```

Collaboration diagram for Ewl_Highlight:



7.71.1 Detailed Description

Inherits from **Ewl_Floater** (p. 671) and extends to provide a highlight widget.

Data Fields

- **Ewl_Floater** floater

7.71.2 Field Documentation

7.71.2.1 Ewl_Floater Ewl_Highlight::floater

Inherit from **Ewl_Floater** (p. 671)

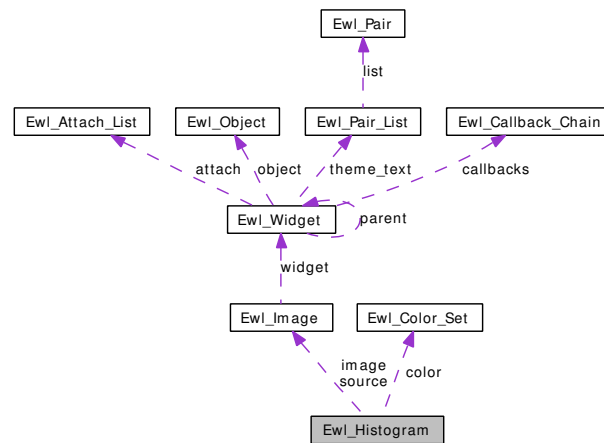
The documentation for this struct was generated from the following file:

- lib/ewl_highlight.h

7.72 Ewl_Histogram Struct Reference

```
#include <ewl_histogram.h>
```

Collaboration diagram for Ewl_Histogram:



7.72.1 Detailed Description

Inherit from the **Ewl_Image** (p. 686) and extends to create a histogram widget.

Data Fields

- **Ewl_Histogram_Channel** channel
- **Ewl_Color_Set** color
- **int graph** [256]
- **Ewl_Image** image
- **int maxv**
- **Ewl_Image * source**

7.72.2 Field Documentation

7.72.2.1 Ewl_Histogram_Channel Ewl_Histogram::channel

Channel of data for graphing

7.72.2.2 Ewl_Color_Set Ewl_Histogram::color

Color for drawing the histogram

7.72.2.3 int Ewl_Histogram::graph[256]

Calculated histogram points

7.72.2.4 Ewl_Image Ewl_Histogram::image

Inherit from the image widget

7.72.2.5 int Ewl_Histogram::maxv

Largest data point in graph

7.72.2.6 Ewl_Image* Ewl_Histogram::source

Source image used for data graph

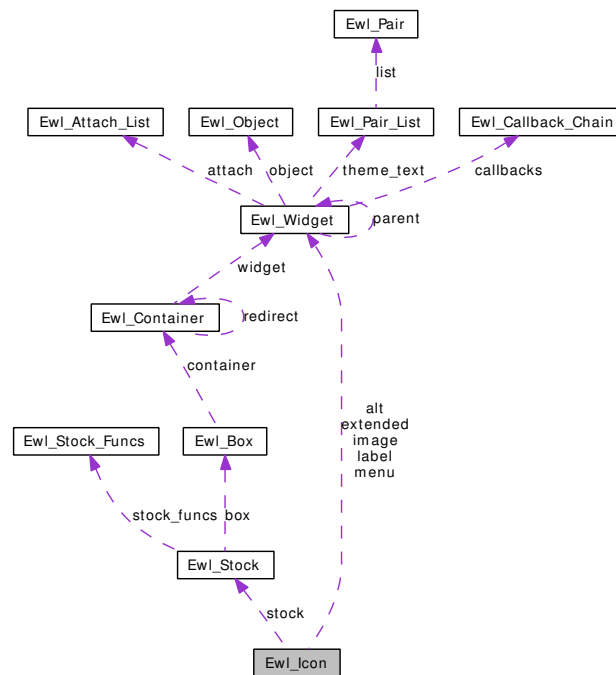
The documentation for this struct was generated from the following file:

- lib/ewl_histogram.h

7.73 Ewl_Icon Struct Reference

```
#include <ewl_icon.h>
```

Collaboration diagram for Ewl_Icon:



7.73.1 Detailed Description

Inherits from **Ewl_Stock** (p. 750) and extends to provide for an icon layout.

Data Fields

- **Ewl_Widget * alt**
- unsigned char **complex_label**:1
- unsigned char **compress_label**:1
- unsigned char **editable**:1
- **Ewl_Widget * extended**
- **Ewl_Icon_Part hidden**
- **Ewl_Widget * image**
- **Ewl_Widget * label**
- char * **label_text**
- **Ewl_Widget * menu**
- **Ewl_Stock stock**
- unsigned char **thumbnailing**:1
- **Ewl_Icon_Type type**

7.73.2 Field Documentation

7.73.2.1 `Ewl_Widget* Ewl_Icon::alt`

The icons alt text

7.73.2.2 `unsigned char Ewl_Icon::complex_label`

Should the label be a `Ewl_Text` (p. 757)?

7.73.2.3 `unsigned char Ewl_Icon::compress_label`

Should the label be compressed?

7.73.2.4 `unsigned char Ewl_Icon::editable`

Is the icon editable?

7.73.2.5 `Ewl_Widget* Ewl_Icon::extended`

The icons extended information

7.73.2.6 `Ewl_Icon_Part Ewl_Icon::hidden`

The hidden part

7.73.2.7 `Ewl_Widget* Ewl_Icon::image`

The icons image

7.73.2.8 `Ewl_Widget* Ewl_Icon::label`

The icons label

7.73.2.9 `char* Ewl_Icon::label_text`

The label text

7.73.2.10 `Ewl_Widget* Ewl_Icon::menu`

The icons menu

7.73.2.11 `Ewl_Stock Ewl_Icon::stock`

Inherit from `Ewl_Box` (p. 586)

7.73.2.12 unsigned char Ewl_Icon::thumbnailing

Should the image be thumbnailed?

7.73.2.13 Ewl_Icon_Type Ewl_Icon::type

The icons type

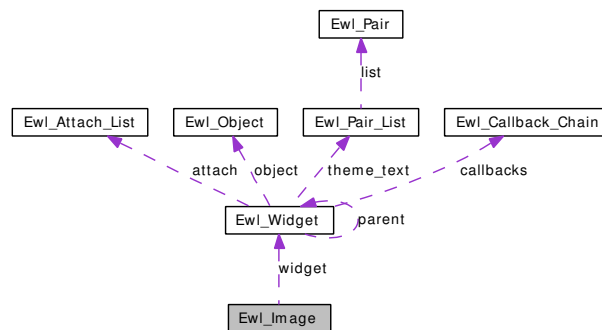
The documentation for this struct was generated from the following file:

- lib/ewl_icon.h

7.74 Ewl_Image Struct Reference

```
#include <ewl_image.h>
```

Collaboration diagram for Ewl_Image:



7.74.1 Detailed Description

Inherits from **Ewl_Widget** (p. 785) and extends to provide an image widget.

Data Fields

- int **ah**
- int **aw**
- int **cs**
- void * **image**
- char * **key**
- int **oh**
- int **ow**
- char * **path**
- char **proportional**
- double **sh**
- double **sw**
- struct {
 - int **h**
 - int **set**
 - int **w**
 - int **x**
 - int **y**
 } **tile**
- Ewl_Image_Type **type**
- Ewl_Widget **widget**

7.74.2 Field Documentation

7.74.2.1 int Ewl_Image::ah

Scale height

7.74.2.2 int Ewl_Image::aw

Scale width

7.74.2.3 int Ewl_Image::cs

Constrain size

7.74.2.4 int Ewl_Image::h

Tile height

7.74.2.5 void* Ewl_Image::image

The evas object for the image

7.74.2.6 char* Ewl_Image::key

The key for the image

7.74.2.7 int Ewl_Image::oh

Original image height

7.74.2.8 int Ewl_Image::ow

Original image width

7.74.2.9 char* Ewl_Image::path

The path to the image

7.74.2.10 char Ewl_Image::proportional

Is the image displayed proportional

7.74.2.11 int Ewl_Image::set

Tiling set?

7.74.2.12 double Ewl_Image::sh

Scale height

7.74.2.13 double Ewl_Image::sw

Scale width

7.74.2.14 `struct { ... } Ewl_Image::tile`

Image tiling information

7.74.2.15 `Ewl_Image_Type Ewl_Image::type`

The type of the image

7.74.2.16 `int Ewl_Image::w`

Tile width

7.74.2.17 `Ewl_Widget Ewl_Image::widget`

Inherit from `Ewl_Widget` (p. 785)

7.74.2.18 `int Ewl_Image::x`

Tile x start

7.74.2.19 `int Ewl_Image::y`

Tile y start

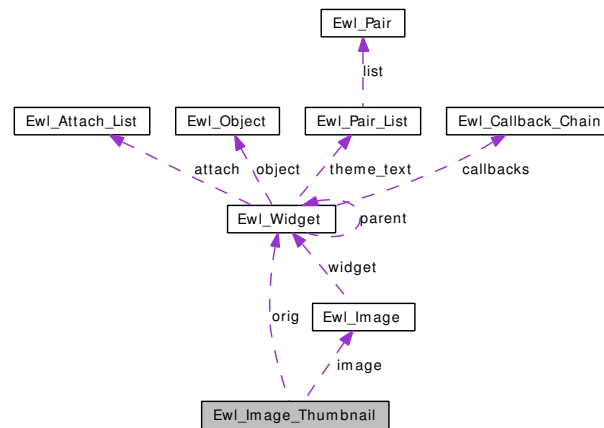
The documentation for this struct was generated from the following file:

- `lib/ewl_image.h`

7.75 Ewl_Image_Thumbnail Struct Reference

```
#include <ewl_image.h>
```

Collaboration diagram for Ewl_Image_Thumbnail:



7.75.1 Detailed Description

Inherits from **Ewl_Image** (p. 686) and extends to provide reference to original image.

Data Fields

- **Ewl_Image** image
- **Ewl_Widget *** orig
- void * thumb

7.75.2 Field Documentation

7.75.2.1 Ewl_Image Ewl_Image_Thumbnail::image

Inherit from **Ewl_Image** (p. 686)

7.75.2.2 Ewl_Widget* Ewl_Image_Thumbnail::orig

Reference to image used to create thumb

7.75.2.3 void* Ewl_Image_Thumbnail::thumb

Outstanding request to thumbnail image

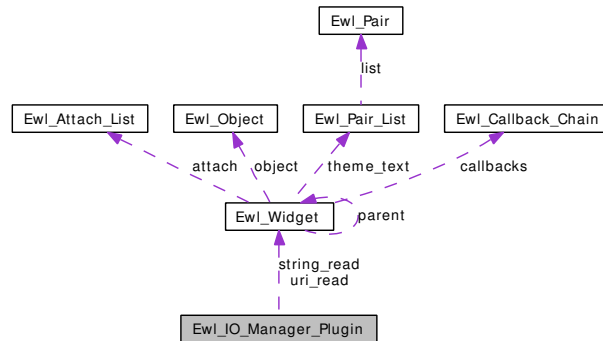
The documentation for this struct was generated from the following file:

- lib/ewl_image.h

7.76 Ewl_IO_Manager_Plugin Struct Reference

```
#include <ewl_io_manager.h>
```

Collaboration diagram for Ewl_IO_Manager_Plugin:



7.76.1 Detailed Description

Provides the base structure for IO Manager plugins.

Data Fields

- void * **handle**
- **Ewl_Widget** *(* **string_read**)(const char *string)
- int(* **string_write**)(Ewl_Widget *data, char **string)
- **Ewl_Widget** *(* **uri_read**)(const char *uri)
- int(* **uri_write**)(Ewl_Widget *data, const char *uri)

7.76.2 Field Documentation

7.76.2.1 void* Ewl_IO_Manager_Plugin::handle

The library handle

7.76.2.2 Ewl_Widget>(* Ewl_IO_Manager_Plugin::string_read)(const char *string)

The string read function

7.76.2.3 int(* Ewl_IO_Manager_Plugin::string_write)(Ewl_Widget *data, char **string)

The string write function

7.76.2.4 Ewl_Widget>(* Ewl_IO_Manager_Plugin::uri_read)(const char *uri)

The URI read fuction

7.76.2.5 `int(* Ewl_IO_Manager_Plugin::uri_write)(Ewl_Widget *data, const char *uri)`

The URI write function

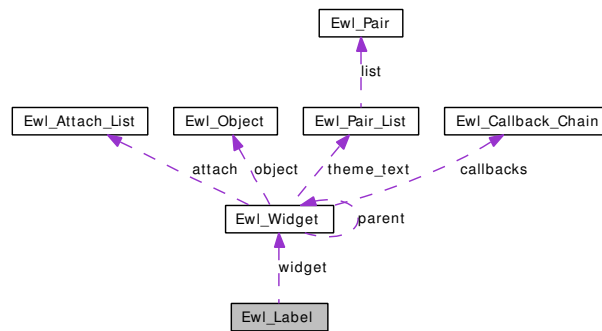
The documentation for this struct was generated from the following file:

- `lib/ewl_io_manager.h`

7.77 Ewl_Label Struct Reference

```
#include <ewl_label.h>
```

Collaboration diagram for Ewl_Label:



7.77.1 Detailed Description

Inherits from **Ewl_Widget** (p. 785) class to provide a simple label.

Data Fields

- **Ewl_Widget** widget

7.77.2 Field Documentation

7.77.2.1 Ewl_Widget Ewl_Label::widget

Inherit from **Ewl_Widget** (p. 785)

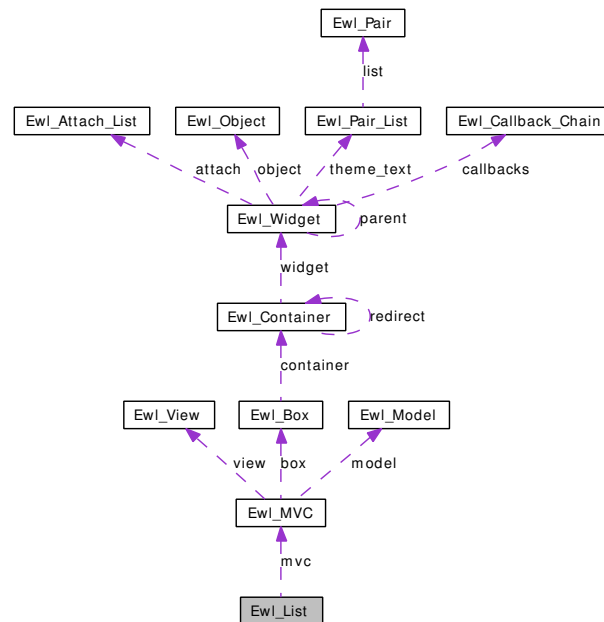
The documentation for this struct was generated from the following file:

- `lib/ewl_label.h`

7.78 Ewl_List Struct Reference

```
#include <ewl_list.h>
```

Collaboration diagram for Ewl_List:



7.78.1 Detailed Description

Inherits from **Ewl_MVC** (p. 705) and extends to provide a list widget.

Data Fields

- **Ewl_MVC mvc**

7.78.2 Field Documentation

7.78.2.1 Ewl_MVC Ewl_List::mvc

The mvc parent

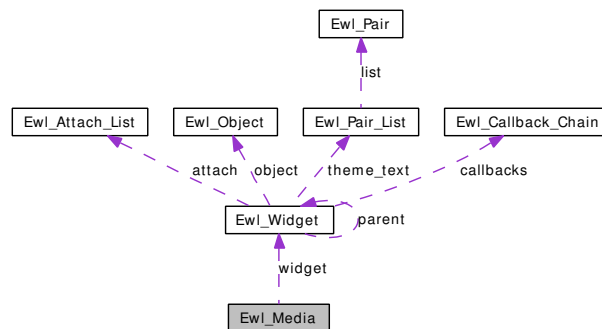
The documentation for this struct was generated from the following file:

- **lib/ewl_list.h**

7.79 Ewl_Media Struct Reference

```
#include <ewl_media.h>
```

Collaboration diagram for Ewl_Media:



7.79.1 Detailed Description

Inherits from the **Ewl_Widget** (p.785) class and extends it to provide for multi-line media layout, obstacle wrapping, and a variety of formatting.

Data Fields

- int **block_seek**
- char * **media**
- **Ewl_Media_Module_Type** module
- unsigned char **mute**:1
- double **position**
- void * **video**
- double **volume**
- **Ewl_Widget** widget

7.79.2 Field Documentation

7.79.2.1 int Ewl_Media::block_seek

Temporarily prevent seeking

7.79.2.2 char* Ewl_Media::media

Path to the media file

7.79.2.3 Ewl_Media_Module_Type Ewl_Media::module

Path to the module file

7.79.2.4 unsigned char Ewl_Media::mute

The mute value

7.79.2.5 double Ewl_Media::position

The position setting

7.79.2.6 void* Ewl_Media::video

Emotion does the actual work

7.79.2.7 double Ewl_Media::volume

The volume setting

7.79.2.8 Ewl_Widget Ewl_Media::widget

Inherit from **Ewl_Widget** (p. 785)

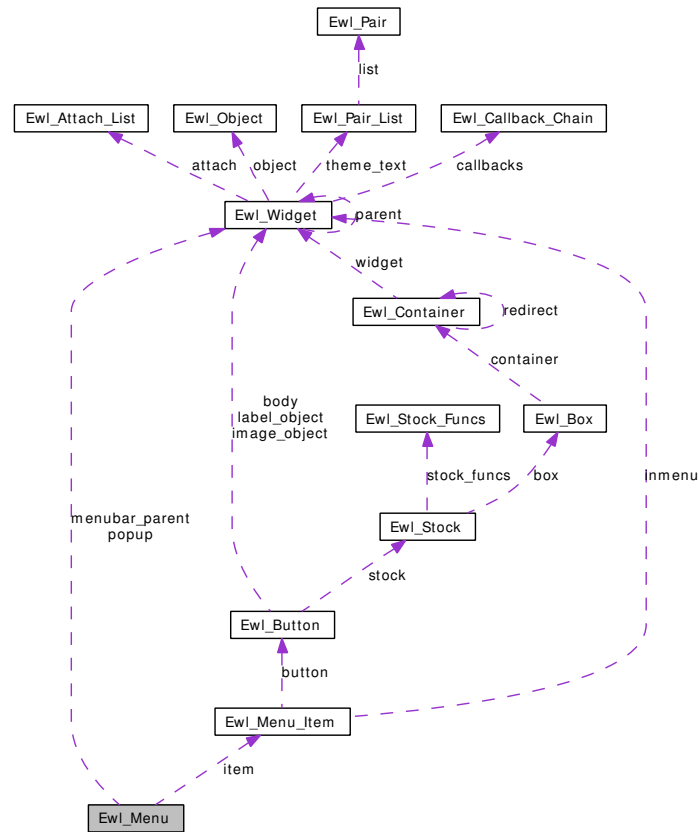
The documentation for this struct was generated from the following file:

- lib/ewl_media.h

7.80 Ewl_Menu Struct Reference

```
#include <ewl_menu.h>
```

Collaboration diagram for Ewl_Menu:



7.80.1 Detailed Description

Inherits from the `Ewl_Menu_Base` and extends to provide a simple menu.

Data Fields

- `Ewl_Menu_Item` `item`
- `Ewl_Widget *` `menubar_parent`
- `Ewl_Widget *` `popup`

7.80.2 Field Documentation

7.80.2.1 `Ewl_Menu_Item` `Ewl_Menu::item`

Inherit from `Ewl_Menu_Base`

7.80.2.2 Ewl_Widget* Ewl_Menu::menubar_parent

Parent menu bar

7.80.2.3 Ewl_Widget* Ewl_Menu::popup

The popup holding the menu items

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

- lib/ewl_menu.h

7.81 Ewl_Menu_Info Struct Reference

```
#include <ewl_menu.h>
```

7.81.1 Detailed Description

Provides a simple structure to contain menu information.

Data Fields

- void(* **cb**)(Ewl_Widget *w, void *ev, void *data)
- char * **img**
- char * **name**

7.81.2 Field Documentation

7.81.2.1 void(* Ewl_Menu_Info::cb)(Ewl_Widget *w, void *ev, void *data)

The menu item callback

7.81.2.2 char* Ewl_Menu_Info::img

The menu item image

7.81.2.3 char* Ewl_Menu_Info::name

The name of the menu item

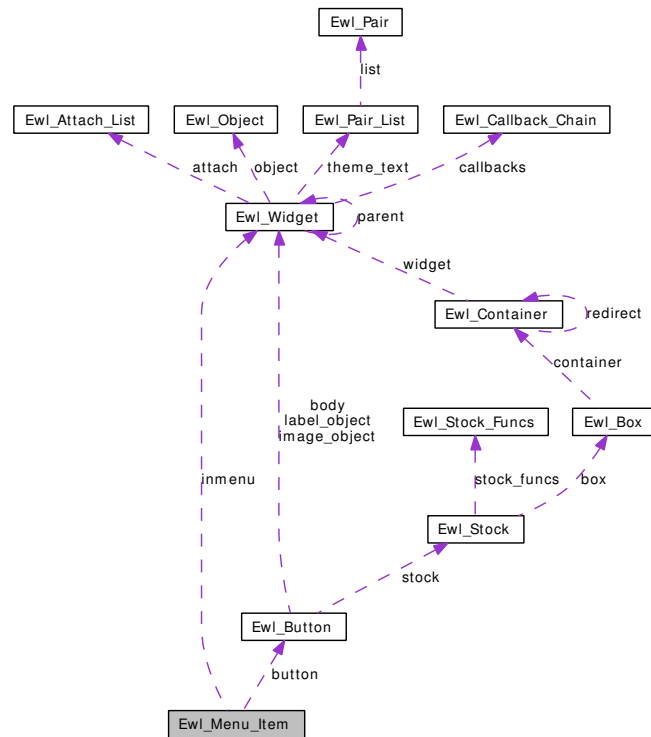
The documentation for this struct was generated from the following file:

- lib/ewl_menu.h

7.82 Ewl_Menu_Item Struct Reference

```
#include <ewl_menu_item.h>
```

Collaboration diagram for Ewl_Menu_Item:



7.82.1 Detailed Description

Inherits from **Ewl_Button** (p. 589) and provides a basic menu item widget.

Data Fields

- **Ewl_Button** button
- **Ewl_Widget *** inmenu

7.82.2 Field Documentation

7.82.2.1 Ewl_Button Ewl_Menu_Item::button

Inherit from **Ewl_Button** (p. 589)

7.82.2.2 Ewl_Widget* Ewl_Menu_Item::inmenu

Set if inside a menu

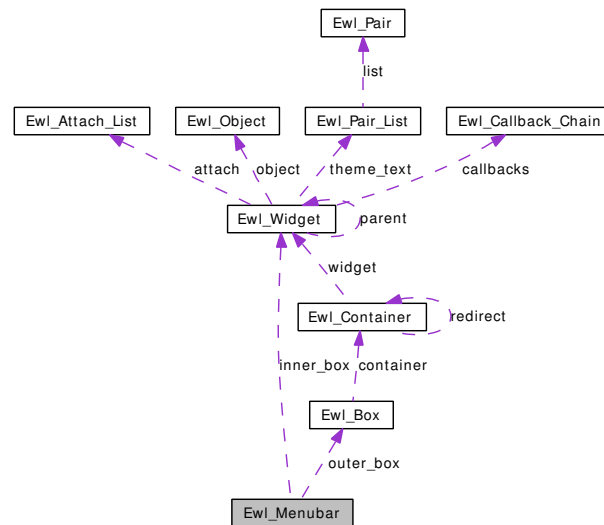
The documentation for this struct was generated from the following file:

- `lib/ewl__menu__item.h`

7.83 Ewl_Menubar Struct Reference

```
#include <ewl_menubar.h>
```

Collaboration diagram for Ewl_Menubar:



7.83.1 Detailed Description

Inherits from the **Ewl_Box** (p. 586) class and extends it to provide for a menubar.

Data Fields

- **Ewl_Widget** * `inner_box`
- **Ewl_Box** `outer_box`

7.83.2 Field Documentation

7.83.2.1 **Ewl_Widget*** **Ewl_Menubar::inner_box**

The box to pack the widgets into

7.83.2.2 **Ewl_Box** **Ewl_Menubar::outer_box**

Inherit from **Ewl_Box** (p. 586)

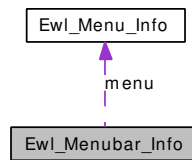
The documentation for this struct was generated from the following file:

- `lib/ewl_menubar.h`

7.84 Ewl_Menubar_Info Struct Reference

```
#include <ewl_menubar.h>
```

Collaboration diagram for Ewl_Menubar_Info:



7.84.1 Detailed Description

Provides a simple menu structure for the menubar.

Data Fields

- **Ewl_Menu_Info * menu**
- **char * name**

7.84.2 Field Documentation

7.84.2.1 Ewl_Menu_Info* Ewl_Menubar_Info::menu

The menu items

7.84.2.2 char* Ewl_Menubar_Info::name

The menu name

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

- **lib/ewl_menubar.h**

7.85 Ewl_Model Struct Reference

```
#include <ewl_model.h>
```

7.85.1 Detailed Description

Holds the callbacks needed to define a model.

Data Fields

- **Ewl_Model_Data_Count** count
- **Ewl_Model_Data_Free** data_free
- **struct** {
 - Ewl_Model_Expansion_Data_Fetch** data
 - Ewl_Model_Data_Expandable** is
 - Ewl_Model_Expansion_Model_Fetch** model
- **Ewl_Model_Data_Fetch** fetch
- **Ewl_Model_Data_Header_Fetch** header
- **Ewl_Model_Data_Sort** sort
- **Ewl_Model_Column_Sortable** sortable

7.85.2 Field Documentation

7.85.2.1 Ewl_Model_Data_Count Ewl_Model::count

Count of data items

7.85.2.2 Ewl_Model_Expansion_Data_Fetch Ewl_Model::data

Get expansion data

7.85.2.3 Ewl_Model_Data_Free Ewl_Model::data_free

Free data passed to view

7.85.2.4 struct { ... } Ewl_Model::expansion

7.85.2.5 Ewl_Model_Data_Fetch Ewl_Model::fetch

Retrieve data for a cell

7.85.2.6 Ewl_Model_Data_Header_Fetch Ewl_Model::header

Retrieve header data

7.85.2.7 Ewl_Model_Data_Expandable Ewl_Model::is

Is the row expandable

7.85.2.8 Ewl_Model_Expansion_Model_Fetch Ewl_Model::model

Get expansion model

7.85.2.9 Ewl_Model_Data_Sort Ewl_Model::sort

Trigger sort on column

7.85.2.10 Ewl_Model_Column_Sortable Ewl_Model::sortable

Is a column sortable

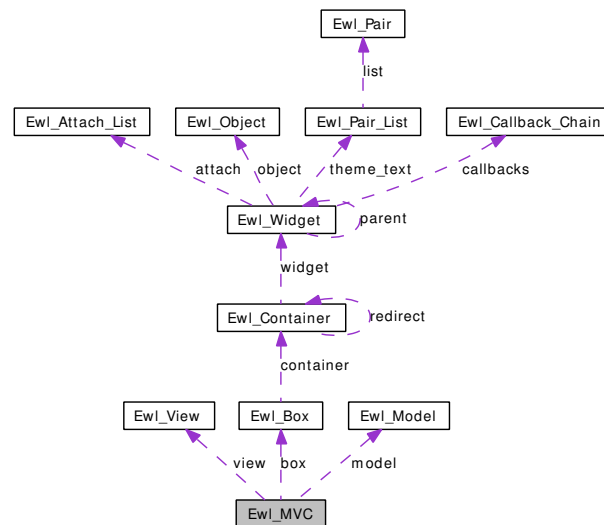
The documentation for this struct was generated from the following file:

- lib/ewl_model.h

7.86 Ewl_MVC Struct Reference

```
#include <ewl_mvc.h>
```

Collaboration diagram for Ewl_MVC:



7.86.1 Detailed Description

Inherits from **Ewl_Box** (p.586) and extends to provide mvc functionality.

Data Fields

- **Ewl_Box** **box**
- struct {
 - void(* **selected_change**)(Ewl_MVC *mvc)
 - void(* **view_change**)(Ewl_MVC *mvc)
 } **cb**
- void * **data**
- unsigned char **dirty**:1
- **Ewl_Model** * **model**
- Ecore_List * **selected**
- **Ewl_Selection_Mode** **selection_mode**
- **Ewl_View** * **view**

7.86.2 Field Documentation

7.86.2.1 Ewl_Box Ewl_MVC::box

Inherit from **Ewl_Box** (p. 586)

7.86.2.2 struct { ... } Ewl_MVC::cb

Callbacks to the inheriting widgets

7.86.2.3 void* Ewl_MVC::data

The mvc data

7.86.2.4 unsigned char Ewl_MVC::dirty

Is the data dirty

7.86.2.5 Ewl_Model* Ewl_MVC::model

The model for the mvc

7.86.2.6 Ecore_List* Ewl_MVC::selected

The selected cells

7.86.2.7 void(* Ewl_MVC::selected_change)(Ewl_MVC *mvc)

Selected change callback

7.86.2.8 Ewl_Selection_Mode Ewl_MVC::selection_mode

The widget selection mode

7.86.2.9 Ewl_View* Ewl_MVC::view

The view for the mvc

7.86.2.10 void(* Ewl_MVC::view_change)(Ewl_MVC *mvc)

View change callback

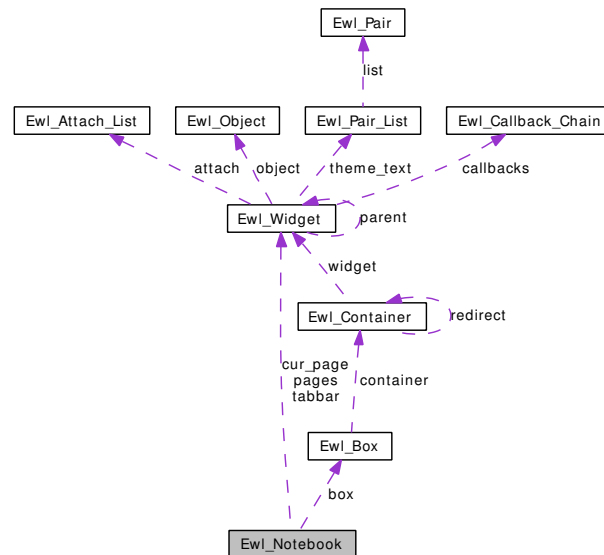
The documentation for this struct was generated from the following file:

- lib/**ewl_mvc.h**

7.87 Ewl_Notebook Struct Reference

```
#include <ewl_notebook.h>
```

Collaboration diagram for Ewl_Notebook:



7.87.1 Detailed Description

Inherits from **Ewl_Box** (p. 586) and extends to provide for a notebook style layout.

Data Fields

- struct {
 - Ewl_Widget * pages
 - Ewl_Widget * tabbar
 } body
- Ewl_Box box
- Ewl_Widget * cur_page
- Ewl_Position tabbar_position

7.87.2 Field Documentation

7.87.2.1 struct { ... } Ewl_Notebook::body

The body of the notebook

7.87.2.2 Ewl_Box Ewl_Notebook::box

Inherit from **Ewl_Box** (p. 586)

7.87.2.3 Ewl_Widget* Ewl_Notebook::cur_page

The currently active page

7.87.2.4 Ewl_Widget* Ewl_Notebook::pages

Holds the notebook pages

7.87.2.5 Ewl_Widget* Ewl_Notebook::tabbar

Holds the tabs

7.87.2.6 Ewl_Position Ewl_Notebook::tabbar_position

The position of the tabbar in the notebook

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

- lib/ewl_notebook.h

7.88 Ewl_Object Struct Reference

```
#include <ewl_object.h>
```

7.88.1 Detailed Description

Provides facilities for sizing, position, alignment and fill policy.

The fields of the object, while their explanations are fairly clear, can be visualized with the following diagram:

The **CURRENT_W(w)** (p. 348) and **CURRENT_H(h)** (p. 348) are macros that provide easy access to the data fields describing the internal area of the **Ewl_Object** (p. 709). While the functions `ewl_object_get_current_w(w)` and `ewl_object_get_current_h(h)` are to access the overall size of the area this **Ewl_Object** (p. 709) resides in. There are corresponding macros and functions for preferred, minimum and maximum sizes as well. There are also functions for setting each of these values.

The affect of the fields when performing layout is as follows:

As illustrated, the fill policy determines how much space an object will use when the request for a specific size is made. When the fill policy contains **EWL_FLAG_FILL_HSHRINK**, **EWL_FLAG_FILL_VSHRINK** or both, the **Ewl_Object** (p. 709) can be resized down to it's minimum size in width, height or both respectively.

The opposite is true for a fill policy containing **EWL_FLAG_FILL_HFILL**, **EWL_FLAG_FILL_VFILL** or both, The **Ewl_Object** (p. 709) will now expand to fill the space up to it's maximum size in the respective direction.

Data Fields

- struct {
 - int **h**
 - int **w**
 - int **x**
 - int **y**
 } **current**
- unsigned int **flags**
- struct {
 - int **b**
 - int **l**
 - int **r**
 - int **t**
 } **insets**
- struct {
 - int **h**
 - int **w**
 } **maximum**
- struct {
 - int **h**
 - int **w**

} minimum

- struct {
 int **b**
 int **l**
 int **r**
 int **t**
} **pad**
- struct {
 int **h**
 int **w**
} **preferred**

7.88.2 Field Documentation

7.88.2.1 int Ewl_Object::b

Bottom value

7.88.2.2 struct { ... } Ewl_Object::current

The current size and position of an object.

7.88.2.3 unsigned int Ewl_Object::flags

Bitmask indicating fill policy and alignment

7.88.2.4 int Ewl_Object::h

Height

7.88.2.5 struct { ... } Ewl_Object::insets

The space padded around the outside of the object. The space inside where children should not be laid out.

7.88.2.6 int Ewl_Object::l

Left value

7.88.2.7 struct { ... } Ewl_Object::maximum

The optimal size of the object in ideal circumstances The guaranteed maximum size this object will receive.

7.88.2.8 struct { ... } Ewl_Object::minimum

The optimal size of the object in ideal circumstances The guaranteed maximum size this object will receive. The guaranteed minimum size this object will receive.

7.88.2.9 struct { ... } Ewl_Object::pad

The space padded around the outside of the object.

7.88.2.10 struct { ... } Ewl_Object::preferred

The optimal size of the object in ideal circumstances

7.88.2.11 int Ewl_Object::r

Right value

7.88.2.12 int Ewl_Object::t

Top value

7.88.2.13 int Ewl_Object::w

Width

7.88.2.14 int Ewl_Object::x

Horizontal position

7.88.2.15 int Ewl_Object::y

Vertical position

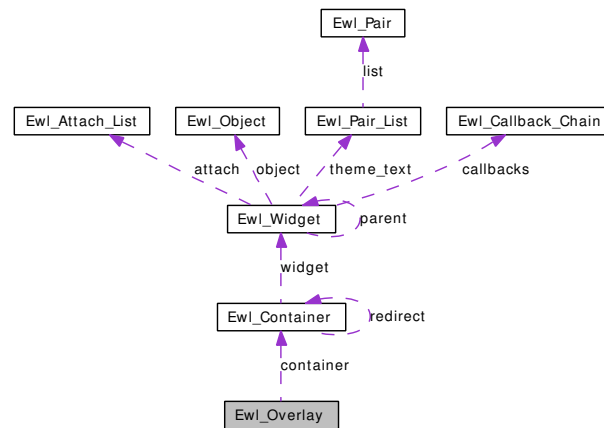
The documentation for this struct was generated from the following file:

- lib/ewl_object.h

7.89 Ewl_Overlay Struct Reference

```
#include <ewl_overlay.h>
```

Collaboration diagram for Ewl_Overlay:



7.89.1 Detailed Description

The class inheriting from **Ewl_Container** (p. 618) that acts as a top level widget for interacting with the evas.

Data Fields

- **Ewl_Container** container

7.89.2 Field Documentation

7.89.2.1 Ewl_Container Ewl_Overlay::container

Inherits from the **Ewl_Container** (p. 618) class

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

- lib/ewl_overlay.h

7.90 Ewl_Pair Struct Reference

```
#include <ewl_widget.h>
```

7.90.1 Detailed Description

Contains a key and a value pair.

Data Fields

- `const char * key`
- `char * value`

7.90.2 Field Documentation

7.90.2.1 `const char* Ewl_Pair::key`

The key

7.90.2.2 `char* Ewl_Pair::value`

The value

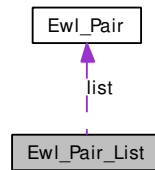
The documentation for this struct was generated from the following file:

- `lib/ewl_widget.h`

7.91 Ewl_Pair_List Struct Reference

```
#include <ewl_widget.h>
```

Collaboration diagram for Ewl_Pair_List:



7.91.1 Detailed Description

Contains a list of key value pairs.

Data Fields

- unsigned int **direct**:1
- unsigned int **len**:31
- **Ewl_Pair ** list**

7.91.2 Field Documentation

7.91.2.1 unsigned int Ewl_Pair_List::direct

Does this list point directly to a pair

7.91.2.2 unsigned int Ewl_Pair_List::len

Length of the list

7.91.2.3 Ewl_Pair** Ewl_Pair_List::list

The key/value list

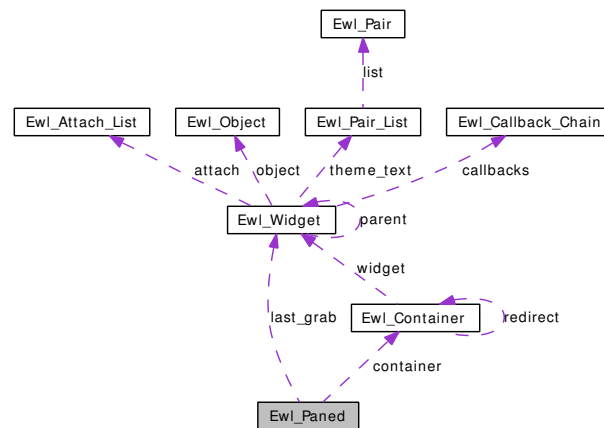
The documentation for this struct was generated from the following file:

- lib/ewl_widget.h

7.92 Ewl_Paned Struct Reference

```
#include <ewl_paned.h>
```

Collaboration diagram for Ewl_Paned:



7.92.1 Detailed Description

Inherits from **Ewl_Container** (p. 618) and extends to provided the paned widget.

Data Fields

- **Ewl_Container** container
- **Ewl_Widget *** last_grab
- int last_pos
- int last_size
- unsigned short new_panes:1
- **Ewl_Orientation** orientation
- unsigned short updating_grabbers:1

7.92.2 Field Documentation

7.92.2.1 Ewl_Container Ewl_Paned::container

Inherit from **Ewl_Container** (p. 618)

7.92.2.2 Ewl_Widget* Ewl_Paned::last_grab

a pointer to the last resized pane

7.92.2.3 int Ewl_Paned::last_pos

the last position

7.92.2.4 int Ewl_Paned::last_size

the last size

7.92.2.5 unsigned short Ewl_Paned::new_panes

are there new visible panes

7.92.2.6 Ewl_Orientation Ewl_Paned::orientation

The orientation of the paned

7.92.2.7 unsigned short Ewl_Paned::updating_grabbers

are we updating the grabbers

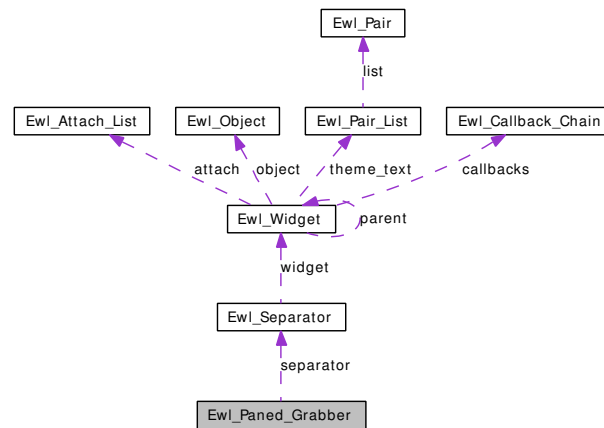
The documentation for this struct was generated from the following file:

- lib/ewl_paned.h

7.93 Ewl_Paned_Grabber Struct Reference

```
#include <ewl_paned.h>
```

Collaboration diagram for Ewl_Paned_Grabber:



7.93.1 Detailed Description

Inherits from **Ewl_Separator** (p. 740) and extends to facilitate the paned grabber.

Data Fields

- unsigned int **placed**:1
- **Ewl_Separator** **separator**

7.93.2 Field Documentation

7.93.2.1 unsigned int Ewl_Paned_Grabber::placed

Has the grabber been placed already?

7.93.2.2 Ewl_Separator Ewl_Paned_Grabber::separator

Inherit from the separator

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

- lib/**ewl_paned.h**

7.94 Ewl_Paned_Layout Struct Reference

7.94.1 Detailed Description

Contains information on a paned layout.

Data Fields

- `int(* current_position)(Ewl_Object *o)`
- `int(* current_size)(Ewl_Object *o)`
- `int(* minimum_size)(Ewl_Object *o)`
- `void(* position_request)(Ewl_Object *o, int pos)`
- `int(* preferred_size)(Ewl_Object *o)`
- `void(* stable_position_request)(Ewl_Object *o, int pos)`
- `void(* stable_request)(Ewl_Object *o, int size)`
- `void(* variable_request)(Ewl_Object *o, int size)`

7.94.2 Field Documentation

7.94.2.1 `int(* Ewl_Paned_Layout::current_position)(Ewl_Object *o)`

7.94.2.2 `int(* Ewl_Paned_Layout::current_size)(Ewl_Object *o)`

7.94.2.3 `int(* Ewl_Paned_Layout::minimum_size)(Ewl_Object *o)`

7.94.2.4 `void(* Ewl_Paned_Layout::position_request)(Ewl_Object *o, int pos)`

7.94.2.5 `int(* Ewl_Paned_Layout::preferred_size)(Ewl_Object *o)`

7.94.2.6 `void(* Ewl_Paned_Layout::stable_position_request)(Ewl_Object *o, int pos)`

7.94.2.7 `void(* Ewl_Paned_Layout::stable_request)(Ewl_Object *o, int size)`

7.94.2.8 `void(* Ewl_Paned_Layout::variable_request)(Ewl_Object *o, int size)`

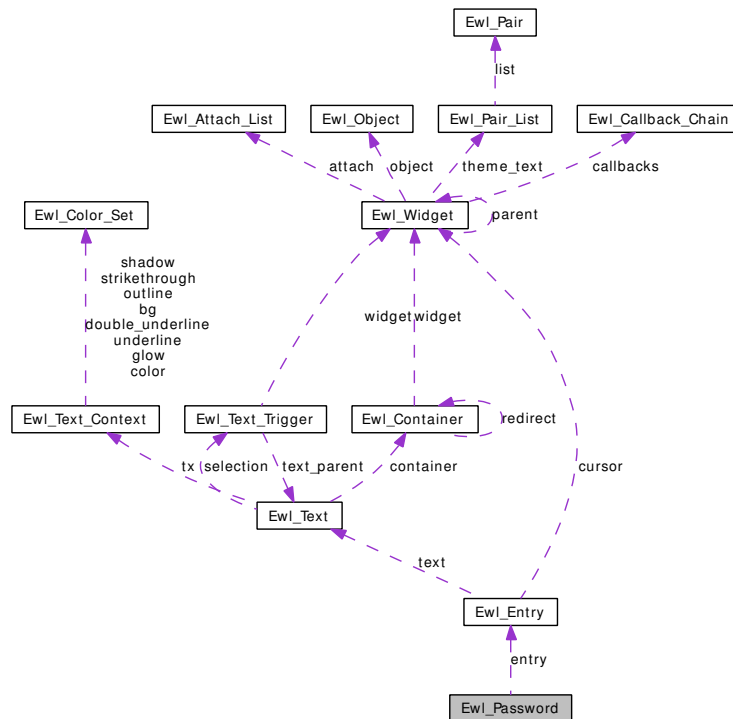
The documentation for this struct was generated from the following file:

- `lib/ewl_paned.c`

7.95 Ewl_Password Struct Reference

```
#include <ewl_password.h>
```

Collaboration diagram for Ewl_Password:



7.95.1 Detailed Description

Inherits from the **Ewl_Entry** (p. 635) and extends it to obscure the text.

Data Fields

- **Ewl_Entry** entry
- char **obscure**
- char * **real_text**

7.95.2 Field Documentation

7.95.2.1 Ewl_Entry Ewl_Password::entry

Inherit from **Ewl_Entry** (p. 635)

7.95.2.2 char Ewl_Password::obscure

Character displayed for password characters

7.95.2.3 char* Ewl_Password::real_text

The password contents

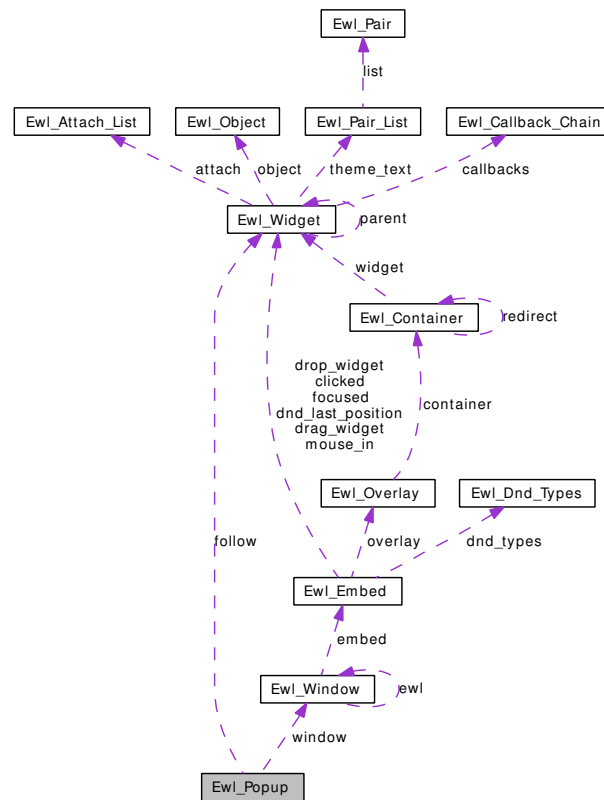
The documentation for this struct was generated from the following file:

- lib/ewl_password.h

7.96 Ewl_Popup Struct Reference

```
#include <ewl_popup.h>
```

Collaboration diagram for Ewl_Popup:



7.96.1 Detailed Description

Inherits from `Ewl_Window` (p. 788) and extends to provide a popup widget.

Data Fields

- `int fit_to_follow`
- `Ewl_Widget * follow`
- struct {
 - `int x`
 - `int y`
 } `mouse`
- struct {
 - `int x`
 - `int y`
 } `offset`
- `Ewl_Popup_Type` type

- **Ewl_Window** window

7.96.2 Field Documentation

7.96.2.1 **int Ewl_Popup::fit_to_follow**

if the popup fits the size to its follow

7.96.2.2 **Ewl_Widget* Ewl_Popup::follow**

The object to follow

7.96.2.3 **struct { ... } Ewl_Popup::mouse**

7.96.2.4 **struct { ... } Ewl_Popup::offset**

7.96.2.5 **Ewl_Popup_Type Ewl_Popup::type**

The Popup type

7.96.2.6 **Ewl_Window Ewl_Popup::window**

Inherit from **Ewl_Window** (p. 788)

7.96.2.7 **int Ewl_Popup::x**

The x position

7.96.2.8 **int Ewl_Popup::y**

The y position

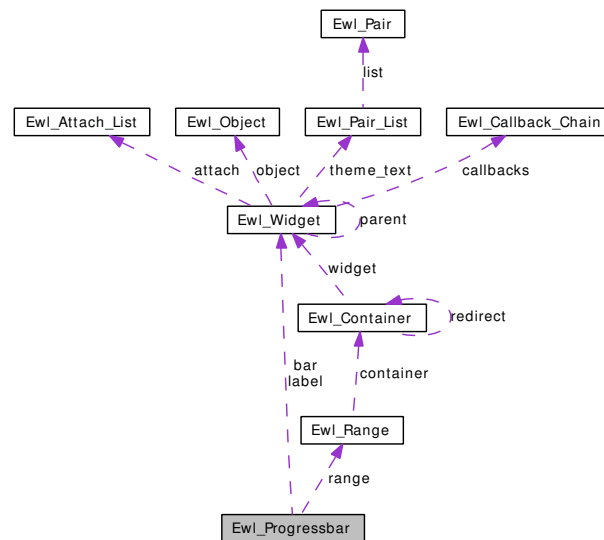
The documentation for this struct was generated from the following file:

- **lib/ewl_popup.h**

7.97 Ewl_Progressbar Struct Reference

```
#include <ewl_progressbar.h>
```

Collaboration diagram for Ewl_Progressbar:



7.97.1 Detailed Description

Inherits from **Ewl_Range** (p. 727) and extends to provide a progressbar widget.

Data Fields

- `int auto_label`
- `Ewl_Widget * bar`
- `Ewl_Widget * label`
- `Ewl_Range range`

7.97.2 Field Documentation

7.97.2.1 `int Ewl_Progressbar::auto_label`

flag if user is setting label or not

7.97.2.2 `Ewl_Widget* Ewl_Progressbar::bar`

The moving bar on top

7.97.2.3 `Ewl_Widget* Ewl_Progressbar::label`

text label on the bar

7.97.2.4 Ewl_Range Ewl_Progressbar::range

Inherit from **Ewl_Range** (p. 727)

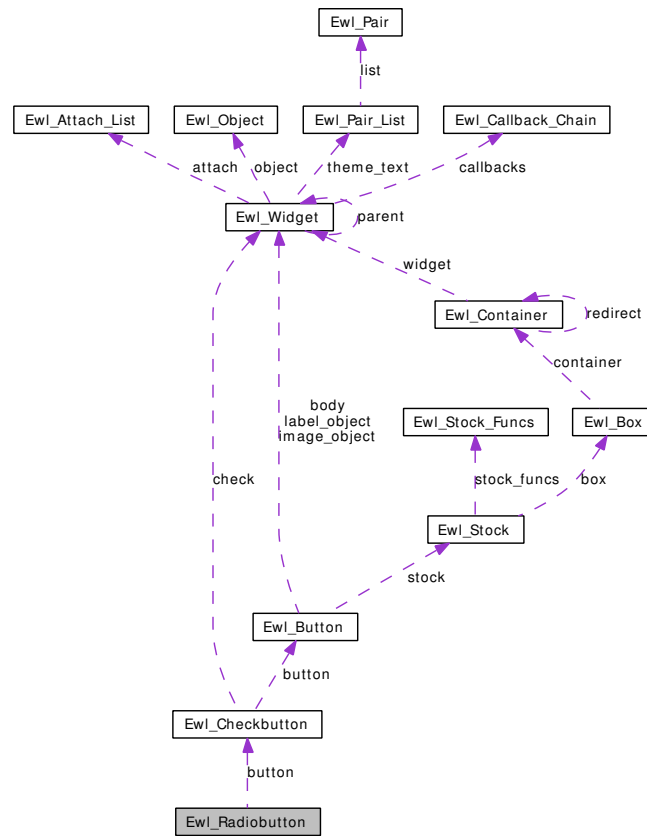
The documentation for this struct was generated from the following file:

- lib/ewl_progressbar.h

7.98 Ewl_Radiobutton Struct Reference

```
#include <ewl_radiobutton.h>
```

Collaboration diagram for Ewl_Radiobutton:



7.98.1 Detailed Description

Inherits from **Ewl_Checkbutton** (p. 598) and extends it to provide grouping buttons to limit to a single selection in a group at a given time.

Data Fields

- **Ewl_Checkbutton** button
- `Ecore_List * chain`

7.98.2 Field Documentation

7.98.2.1 Ewl_Checkbutton Ewl_Radiobutton::button

Inherit from **Ewl_Checkbutton** (p. 598)

7.98.2.2 Ecore_List* Ewl_Radiobutton::chain

List of members of the group

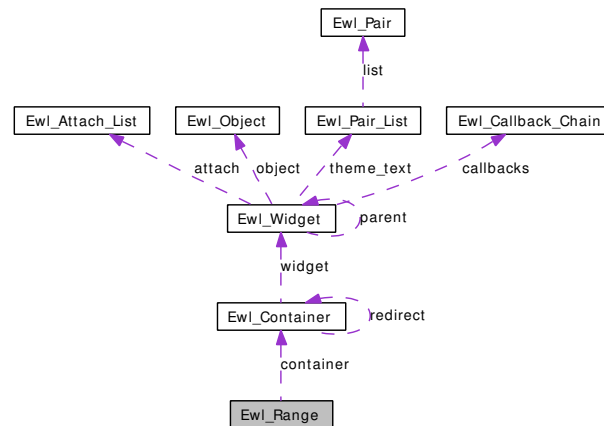
The documentation for this struct was generated from the following file:

- lib/ewl_radiobutton.h

7.99 Ewl_Range Struct Reference

```
#include <ewl_range.h>
```

Collaboration diagram for Ewl_Range:



7.99.1 Detailed Description

Inherits from **Ewl_Container** (p. 618) and extends to provide a basic range widget.

Data Fields

- **Ewl_Container** container
- int invert
- double max_val
- double min_val
- double step
- double value

7.99.2 Field Documentation

7.99.2.1 Ewl_Container Ewl_Range::container

Inherit from **Ewl_Container** (p. 618)

7.99.2.2 int Ewl_Range::invert

Invert the axis

7.99.2.3 double Ewl_Range::max_val

Maximal valide value

7.99.2.4 double Ewl_Range::min_val

Minimal valide value

7.99.2.5 double Ewl_Range::step

Size of increments in the range

7.99.2.6 double Ewl_Range::value

Currently chosen value

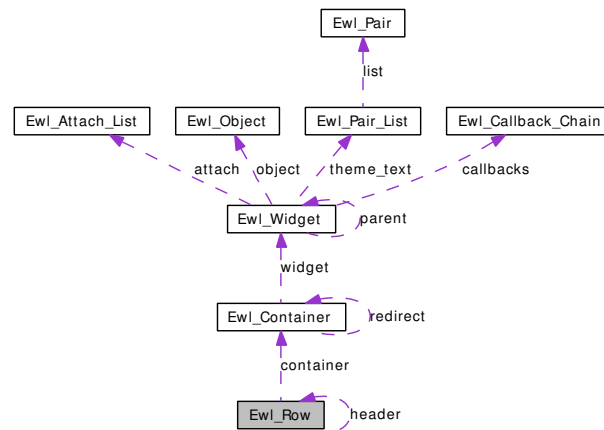
The documentation for this struct was generated from the following file:

- lib/ewl_range.h

7.100 Ewl_Row Struct Reference

```
#include <ewl_row.h>
```

Collaboration diagram for Ewl_Row:



7.100.1 Detailed Description

Inherits from **Ewl_Container** (p. 618) to create a tree row that lays out cells in a left to right fashion.

Data Fields

- **Ewl_Container** container
- **Ewl_Row *** header

7.100.2 Field Documentation

7.100.2.1 Ewl_Container Ewl_Row::container

Inherit from **Ewl_Container** (p. 618)

7.100.2.2 Ewl_Row* Ewl_Row::header

The header row

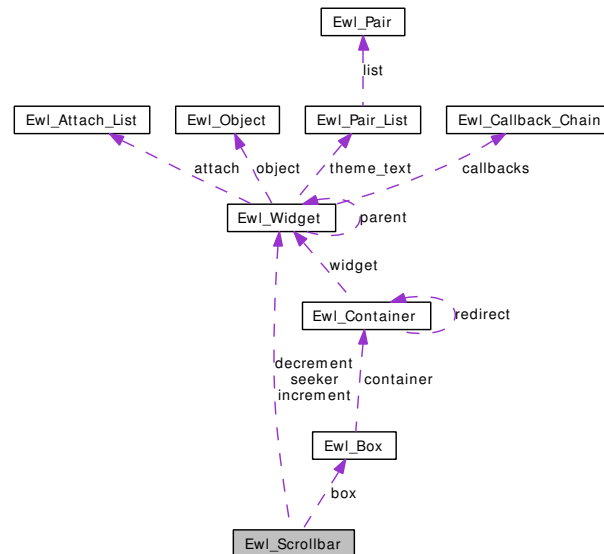
The documentation for this struct was generated from the following file:

- **lib/ewl_row.h**

7.101 Ewl_Scrollbar Struct Reference

```
#include <ewl_scrollbar.h>
```

Collaboration diagram for Ewl_Scrollbar:



7.101.1 Detailed Description

Inherits from **Ewl_Box** (p.586) and extends to provide a scrollbar widget.

Data Fields

- **Ewl_Box** **box**
- unsigned int **buttons_alignment**
- **Ewl_Widget** * **decrement**
- signed char **direction**
- double **fill_percentage**
- **Ewl_Widget** * **increment**
- signed char **invert**
- **Ewl_Widget** * **seeker**
- double **start_time**
- **Ecore_Timer** * **timer**

7.101.2 Field Documentation

7.101.2.1 Ewl_Box Ewl_Scrollbar::box

Inherit from **Ewl_Box** (p.586)

7.101.2.2 unsigned int Ewl_Scrollbar::buttons_alignment

The ordering of buttons

7.101.2.3 Ewl_Widget* Ewl_Scrollbar::decrement

The internal decrement button

7.101.2.4 signed char Ewl_Scrollbar::direction

Direction for the scrollbar

7.101.2.5 double Ewl_Scrollbar::fill_percentage

Ratio of size for draggable

7.101.2.6 Ewl_Widget* Ewl_Scrollbar::increment

The internal increment button

7.101.2.7 signed char Ewl_Scrollbar::invert

Invert the scrolling direction

7.101.2.8 Ewl_Widget* Ewl_Scrollbar::seeker

The internal **Ewl_Seeker** (p. 734)

7.101.2.9 double Ewl_Scrollbar::start_time

Time scrolling began

7.101.2.10 Ecore_Timer* Ewl_Scrollbar::timer

Repeating timer for scrolling

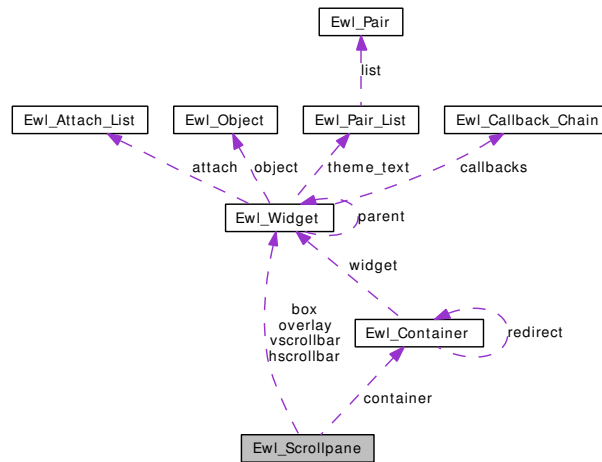
The documentation for this struct was generated from the following file:

- lib/ewl_scrollbar.h

7.102 Ewl_Scrollpane Struct Reference

```
#include <ewl_scrollpane.h>
```

Collaboration diagram for Ewl_Scrollpane:



7.102.1 Detailed Description

Inherits from **Ewl_Container** (p. 618) and extends it to enclose a single widget in vertical and horizontal scrollbars for panning around displaying different regions of the enclosed widget.

Data Fields

- **Ewl_Widget * box**
- **Ewl_Container container**
- **Ewl_Scrollpane_Flags hflag**
- **Ewl_Widget * hscrollbar**
- **Ewl_Widget * overlay**
- **Ewl_Scrollpane_Flags vflag**
- **Ewl_Widget * vscrollbar**

7.102.2 Field Documentation

7.102.2.1 Ewl_Widget* Ewl_Scrollpane::box

Lays out enclosed widget

7.102.2.2 Ewl_Container Ewl_Scrollpane::container

Inherit from **Ewl_Container** (p. 618)

7.102.2.3 Ewl_Scrollpane_Flags Ewl_Scrollpane::hflag

Flags for horizontal scrollbar

7.102.2.4 Ewl_Widget* Ewl_Scrollpane::hscrollbar

Horizontal scrollbar

7.102.2.5 Ewl_Widget* Ewl_Scrollpane::overlay

Clips the enclosed widget

7.102.2.6 Ewl_Scrollpane_Flags Ewl_Scrollpane::vflag

Flags for vertical scrollbar

7.102.2.7 Ewl_Widget* Ewl_Scrollpane::vscrollbar

Vertical scrollbar

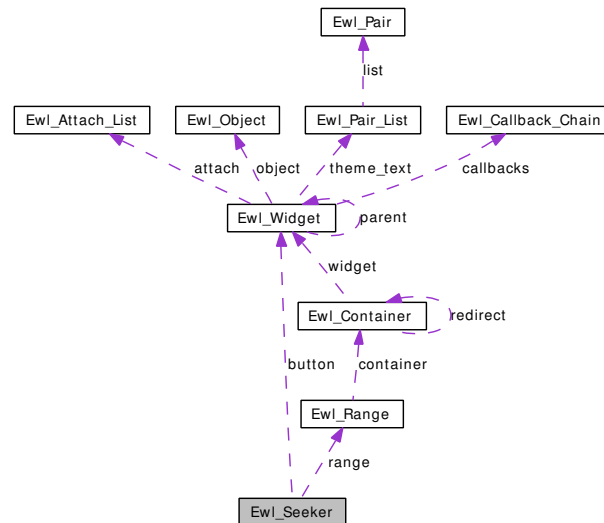
The documentation for this struct was generated from the following file:

- lib/ewl_scrollpane.h

7.103 Ewl_Seeker Struct Reference

```
#include <ewl_seeker.h>
```

Collaboration diagram for Ewl_Seeker:



7.103.1 Detailed Description

Inherits from **Ewl_Range** (p. 727) and extends to provide a seeker widget.

Data Fields

- **int** **autohide**
- **Ewl_Widget *** **button**
- **int** **dragstart**
- **Ewl_Orientation** **orientation**
- **Ewl_Range** **range**
- **double** **start_time**
- **Ecore_Timer *** **timer**

7.103.2 Field Documentation

7.103.2.1 **int** **Ewl_Seeker::autohide**

Indicator to hide when not scrollable

7.103.2.2 **Ewl_Widget*** **Ewl_Seeker::button**

Draggable widget for selecting value

7.103.2.3 int Ewl_Seeker::dragstart

The coordinate where the drag starts

7.103.2.4 Ewl_Orientation Ewl_Seeker::orientation

Indicates layout direction

7.103.2.5 Ewl_Range Ewl_Seeker::range

Inherit from **Ewl_Range** (p. 727)

7.103.2.6 double Ewl_Seeker::start_time

Time the timer was started

7.103.2.7 Ecore_Timer* Ewl_Seeker::timer

Timer for scroll repeating

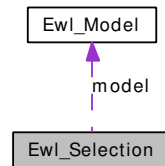
The documentation for this struct was generated from the following file:

- lib/ewl_seeker.h

7.104 Ewl_Selection Struct Reference

```
#include <ewl_mvc.h>
```

Collaboration diagram for Ewl_Selection:



7.104.1 Detailed Description

Structure to store information on selections.

Data Fields

- void * **data**
- void * **highlight**
- Ewl_Model * **model**
- Ewl_Selection_Type **type**

7.104.2 Field Documentation

7.104.2.1 void* Ewl_Selection::data

Data containing the selection

7.104.2.2 void* Ewl_Selection::highlight

highlight widgets for the MVC

7.104.2.3 Ewl_Model* Ewl_Selection::model

The model to work with this selection

7.104.2.4 Ewl_Selection_Type Ewl_Selection::type

The type of selection

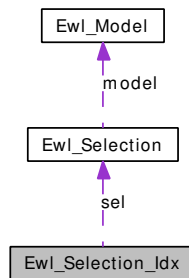
The documentation for this struct was generated from the following file:

- lib/**ewl_mvc.h**

7.105 Ewl_Selection_Idx Struct Reference

```
#include <ewl_mvc.h>
```

Collaboration diagram for Ewl_Selection_Idx:



7.105.1 Detailed Description

Structure to store information on a single index selection.

Data Fields

- `int column`
- `int row`
- `Ewl_Selection sel`

7.105.2 Field Documentation

7.105.2.1 `int Ewl_Selection_Idx::column`

Index column

7.105.2.2 `int Ewl_Selection_Idx::row`

Index row

7.105.2.3 `Ewl_Selection Ewl_Selection_Idx::sel`

Inherit from **Ewl_Selection** (p. 736)

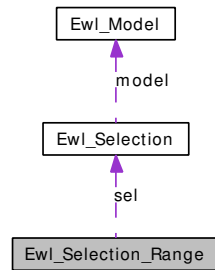
The documentation for this struct was generated from the following file:

- `lib/ewl_mvc.h`

7.106 Ewl_Selection_Range Struct Reference

```
#include <ewl_mvc.h>
```

Collaboration diagram for Ewl_Selection_Range:



7.106.1 Detailed Description

Structure to store a range of selected cells.

Data Fields

- struct {
 int **column**
 int **row**
} **end**
- **Ewl_Selection sel**
- struct {
 int **column**
 int **row**
} **start**

7.106.2 Field Documentation

7.106.2.1 int Ewl_Selection_Range::column

Index column

7.106.2.2 struct { ... } Ewl_Selection_Range::end

Start of range End of range

7.106.2.3 int Ewl_Selection_Range::row

Index row

7.106.2.4 Ewl_Selection Ewl_Selection_Range::sel

Inherit from **Ewl_Selection** (p. 736)

7.106.2.5 struct { ... } Ewl_Selection_Range::start

Start of range

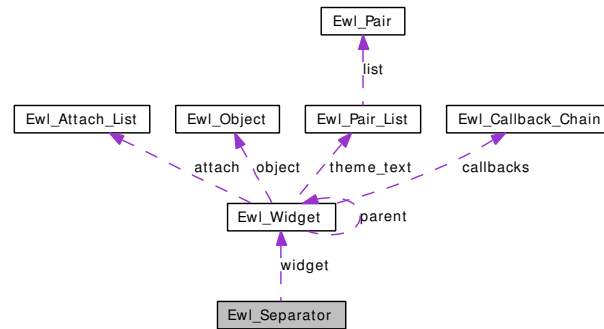
The documentation for this struct was generated from the following file:

- lib/ewl_mvc.h

7.107 Ewl_Separator Struct Reference

```
#include <ewl_separator.h>
```

Collaboration diagram for Ewl_Separator:



7.107.1 Detailed Description

Inherits from **Ewl_Widget** (p. 785) to allow drawing a horizontal or vertical line between widgets, depending on the orientation the **Ewl_Separator** (p. 740).

Data Fields

- **Ewl_Orientation** orientation
- **Ewl_Widget** widget

7.107.2 Field Documentation

7.107.2.1 Ewl_Orientation Ewl_Separator::orientation

Sets drawing horizontal or vertical

7.107.2.2 Ewl_Widget Ewl_Separator::widget

Inherit from **Ewl_Widget** (p. 785)

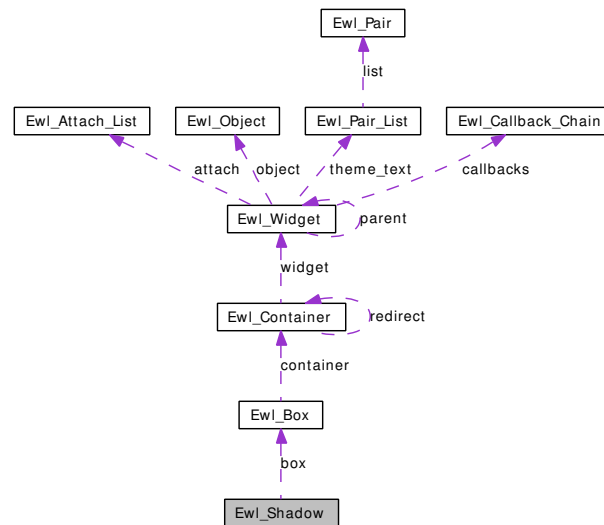
The documentation for this struct was generated from the following file:

- lib/ewl_separator.h

7.108 Ewl_Shadow Struct Reference

```
#include <ewl_shadow.h>
```

Collaboration diagram for Ewl_Shadow:



7.108.1 Detailed Description

Inherits from **Ewl_Box** (p. 586) to allow drawing a border and shadow decoration around widgets.

Data Fields

- **Ewl_Box** box

7.108.2 Field Documentation

7.108.2.1 Ewl_Box Ewl_Shadow::box

Inherit from **Ewl_Box** (p. 586)

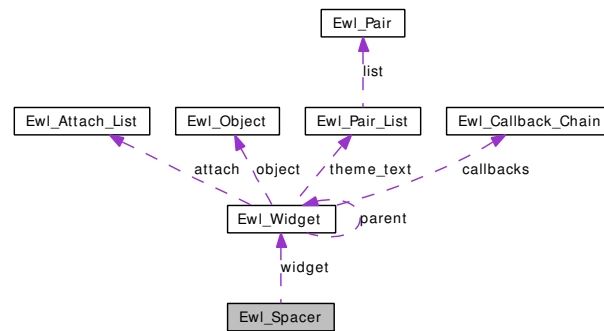
The documentation for this struct was generated from the following file:

- lib/ewl_shadow.h

7.109 Ewl_Spacer Struct Reference

```
#include <ewl_spacer.h>
```

Collaboration diagram for Ewl_Spacer:



7.109.1 Detailed Description

Inherits from **Ewl_Widget** (p. 785) and provides as a spacer between widgets.

Data Fields

- **Ewl_Widget** widget

7.109.2 Field Documentation

7.109.2.1 Ewl_Widget Ewl_Spacer::widget

Inherit from **Ewl_Widget** (p. 785)

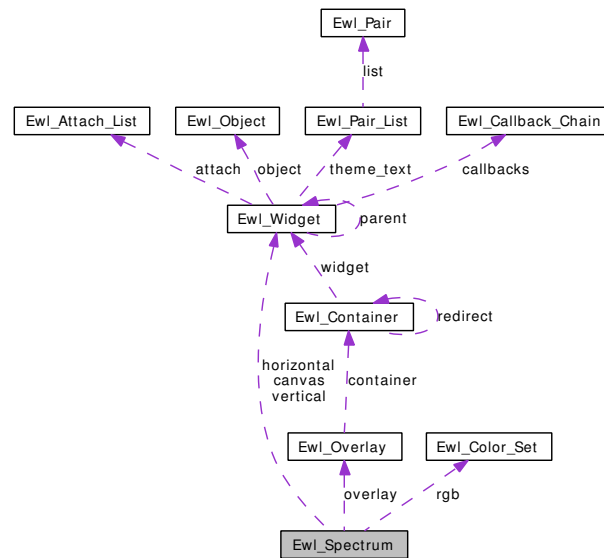
The documentation for this struct was generated from the following file:

- lib/**ewl_spacer.h**

7.110 Ewl_Spectrum Struct Reference

```
#include <ewl_spectrum.h>
```

Collaboration diagram for Ewl_Spectrum:



7.110.1 Detailed Description

Inherits from **Ewl_Overlay** (p. 712) and extends to provide a colour spectrum.

Data Fields

- **Ewl_Widget * canvas**
- struct {
 - Ewl_Widget * horizontal**
 - Ewl_Widget * vertical**
 } **cross_hairs**
- unsigned int **dirty:1**
- struct {
 - double **h**
 - double **s**
 - double **v**
 } **hsv**
- **Ewl_Color_Mode mode**
- **Ewl_Overlay overlay**
- **Ewl_Color_Set rgb**
- **Ewl_Spectrum_Type type**

7.110.2 Field Documentation

7.110.2.1 `Ewl_Widget* Ewl_Spectrum::canvas`

The canvas to draw on

7.110.2.2 `struct { ... } Ewl_Spectrum::cross_hairs`

Selection crosshairs

7.110.2.3 `unsigned int Ewl_Spectrum::dirty`

The flag to queue redraw

7.110.2.4 `double Ewl_Spectrum::h`

The current hue

7.110.2.5 `Ewl_Widget* Ewl_Spectrum::horizontal`

Horizontal portion of the crosshairs

7.110.2.6 `struct { ... } Ewl_Spectrum::hsv`

The HSV setting of the spectrum

7.110.2.7 `Ewl_Color_Mode Ewl_Spectrum::mode`

The mode of the spectrum

7.110.2.8 `Ewl_Overlay Ewl_Spectrum::overlay`

Inherit from `Ewl_Overlay` (p. 712)

7.110.2.9 `Ewl_Color_Set Ewl_Spectrum::rgb`

Current RBG colour of the spectrum

7.110.2.10 `double Ewl_Spectrum::s`

The current saturation

7.110.2.11 `Ewl_Spectrum_Type Ewl_Spectrum::type`

The type of the spectrum

7.110.2.12 double Ewl_Spectrum::v

The current value

7.110.2.13 Ewl_Widget* Ewl_Spectrum::vertical

Vertical portion of the crosshairs

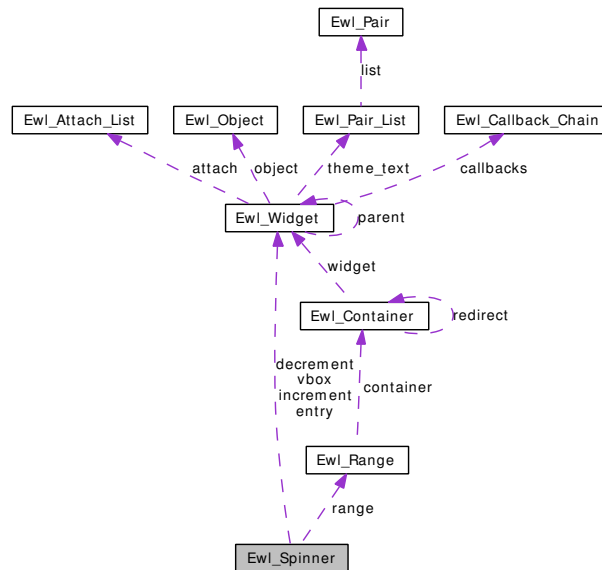
The documentation for this struct was generated from the following file:

- lib/ewl_spectrum.h

7.111 Ewl_Spinner Struct Reference

```
#include <ewl_spinner.h>
```

Collaboration diagram for Ewl_Spinner:



7.111.1 Detailed Description

Inherits from **Ewl_Range** (p. 727) and extends to provide a spinner widget.

Data Fields

- **Ewl_Widget * decrement**
- unsigned char **digits**
- int **direction**
- **Ewl_Widget * entry**
- **Ewl_Widget * increment**
- double **last_value**
- **Ewl_Range range**
- double **start_time**
- **Ecore_Timer * timer**
- **Ewl_Widget * vbox**

7.111.2 Field Documentation

7.111.2.1 Ewl_Widget* Ewl_Spinner::decrement

Ewl_Button (p. 589) to subtract value

7.111.2.2 unsigned char Ewl_Spinner::digits

Number of digits displayed after decimal

7.111.2.3 int Ewl_Spinner::direction

Indicate increasing/decreasing value

7.111.2.4 Ewl_Widget* Ewl_Spinner::entry

The **Ewl_Entry** (p. 635) displaying value

7.111.2.5 Ewl_Widget* Ewl_Spinner::increment

Ewl_Button (p. 589) to add value

7.111.2.6 double Ewl_Spinner::last_value

The last value while spinning

7.111.2.7 Ewl_Range Ewl_Spinner::range

Inherit from **Ewl_Range** (p. 727)

7.111.2.8 double Ewl_Spinner::start_time

Time the spinner was pressed

7.111.2.9 Ecore_Timer* Ewl_Spinner::timer

Timer for tracking mouse button held down

7.111.2.10 Ewl_Widget* Ewl_Spinner::vbox

Ewl_Box (p. 586) to hold the buttons

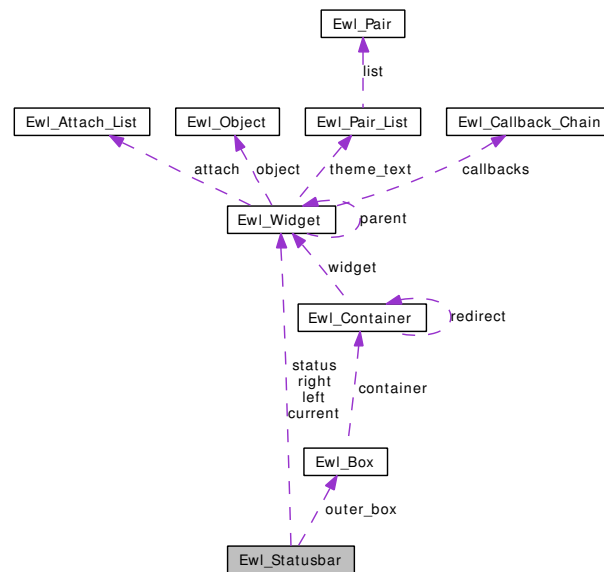
The documentation for this struct was generated from the following file:

- lib/ewl_spinner.h

7.112 Ewl_Statusbar Struct Reference

```
#include <ewl_statusbar.h>
```

Collaboration diagram for Ewl_Statusbar:



7.112.1 Detailed Description

Inherits from the **Ewl_Box** (p. 586) class and extends it to provide for a simple status bar.

Data Fields

- **Ewl_Widget * current**
- **Ewl_Widget * left**
- **Ewl_Box outer_box**
- **Ewl_Widget * right**
- **Ecore_List * stack**
- **Ewl_Widget * status**

7.112.2 Field Documentation

7.112.2.1 Ewl_Widget* Ewl_Statusbar::current

The currently displayed status

7.112.2.2 Ewl_Widget* Ewl_Statusbar::left

The left container

7.112.2.3 Ewl_Box Ewl_Statusbar::outer_box

Inherit from **Ewl_Box** (p. 586)

7.112.2.4 Ewl_Widget* Ewl_Statusbar::right

The right container

7.112.2.5 Ecore_List* Ewl_Statusbar::stack

The stack of status'

7.112.2.6 Ewl_Widget* Ewl_Statusbar::status

The status container

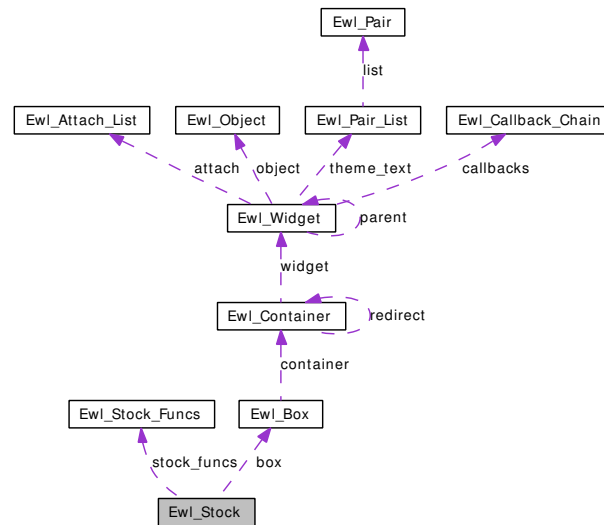
The documentation for this struct was generated from the following file:

- lib/ewl_statusbar.h

7.113 Ewl_Stock Struct Reference

```
#include <ewl_stock.h>
```

Collaboration diagram for Ewl_Stock:



7.113.1 Detailed Description

Inherits from **Ewl_Box** (p.586) and provides the base for the other stock icons.

Data Fields

- **Ewl_Box** box
- **Ewl_Stock_Funcs** * stock_funcs
- **Ewl_Stock_Type** stock_type

7.113.2 Field Documentation

7.113.2.1 Ewl_Box Ewl_Stock::box

Inherit from the box for adding widgets

7.113.2.2 Ewl_Stock_Funcs* Ewl_Stock::stock_funcs

The stock functions

7.113.2.3 Ewl_Stock_Type Ewl_Stock::stock_type

The stock type of the stock

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

- lib/ewl__stock.h

7.114 Ewl_Stock_Funcs Struct Reference

```
#include <ewl_stock.h>
```

7.114.1 Detailed Description

The function callbacks that an inheriting widget needs to implement.

Data Fields

- `Ewl_Stock_Image_Set image_set`
- `Ewl_Stock_Label_Set label_set`
- `Ewl_Stock_Tooltip_Set tooltip_set`

7.114.2 Field Documentation

7.114.2.1 `Ewl_Stock_Image_Set Ewl_Stock_Funcs::image_set`

The image set function

7.114.2.2 `Ewl_Stock_Label_Set Ewl_Stock_Funcs::label_set`

The label set function

7.114.2.3 `Ewl_Stock_Tooltip_Set Ewl_Stock_Funcs::tooltip_set`

The tooltip set function

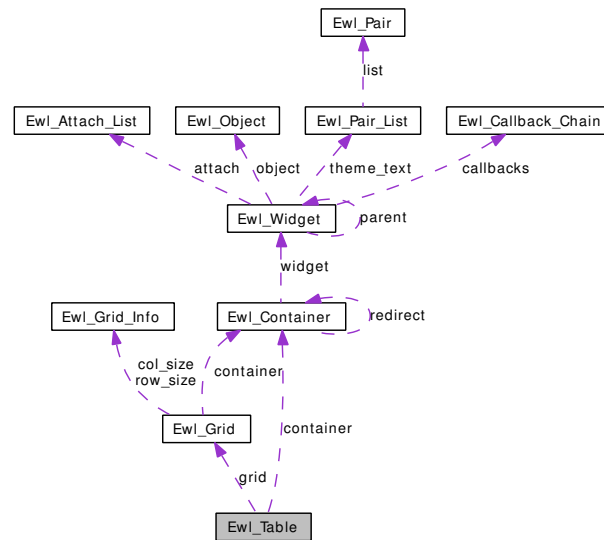
The documentation for this struct was generated from the following file:

- `lib/ewl_stock.h`

7.115 Ewl_Table Struct Reference

```
#include <ewl_table.h>
```

Collaboration diagram for Ewl_Table:



7.115.1 Detailed Description

Inherits from **Ewl_Container** (p.618) and extends to provide a table layout.

Data Fields

- char ** col_headers
- Ewl_Container container
- Ewl_Grid * grid
- unsigned int homogeneous_h
- unsigned int homogeneous_v
- int row_select
- struct {
 - int end_c
 - int end_r
 - int start_c
 - int start_r
 } selected

7.115.2 Field Documentation

7.115.2.1 char** Ewl_Table::col_headers

The column headers

7.115.2.2 Ewl_Container Ewl_Table::container

Inherit from **Ewl_Container** (p. 618)

7.115.2.3 int Ewl_Table::end_c

Selection end column

7.115.2.4 int Ewl_Table::end_r

Selection end row

7.115.2.5 Ewl_Grid* Ewl_Table::grid

Table uses a grid based layout

7.115.2.6 unsigned int Ewl_Table::homogeneous_h

Horizontal homogeneous flag

7.115.2.7 unsigned int Ewl_Table::homogeneous_v

Vertical homogeneous flag

7.115.2.8 int Ewl_Table::row_select

boolean: select entire rows

7.115.2.9 struct { ... } Ewl_Table::selected

Currently selected rows

7.115.2.10 int Ewl_Table::start_c

Selection start column

7.115.2.11 int Ewl_Table::start_r

Selection start row

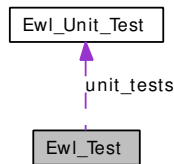
The documentation for this struct was generated from the following file:

- lib/ewl_table.h

7.116 Ewl_Test Struct Reference

```
#include <Ewl_Test.h>
```

Collaboration diagram for Ewl_Test:



7.116.1 Detailed Description

Holds information on an Ewl Test set.

Data Fields

- `const char * filename`
- `int(* func)(Ewl_Container *con)`
- `void * handle`
- `const char * name`
- `const char * tip`
- `Ewl_Test_Type type`
- `Ewl_Unit_Test * unit_tests`

7.116.2 Field Documentation

7.116.2.1 `const char* Ewl_Test::filename`

Filename of the test

7.116.2.2 `int(* Ewl_Test::func)(Ewl_Container *con)`

The test function

7.116.2.3 `void* Ewl_Test::handle`

Dynamic file handle for test

7.116.2.4 `const char* Ewl_Test::name`

The test name

7.116.2.5 `const char* Ewl_Test::tip`

Tooltip for the test

7.116.2.6 Ewl_Test_Type Ewl_Test::type

The type of test

7.116.2.7 Ewl_Unit_Test* Ewl_Test::unit_tests

The test unit tests

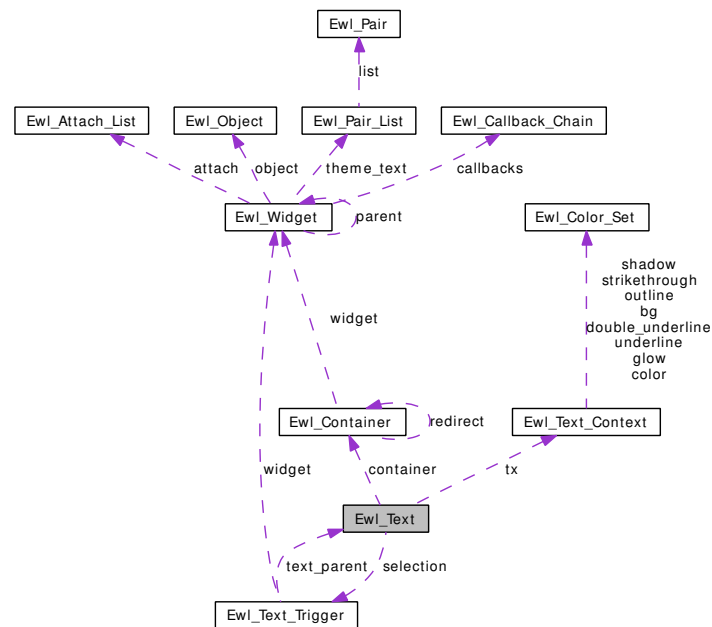
The documentation for this struct was generated from the following file:

- bin/**Ewl_Test.h**

7.117 Ewl_Text Struct Reference

```
#include <ewl_text.h>
```

Collaboration diagram for Ewl_Text:



7.117.1 Detailed Description

Inherits from the **Ewl_Container** (p.618) class and extends it to provide text layout and formatting.

Data Fields

- **Ewl_Container** container
- unsigned int **cursor_position**
- unsigned char **delete_count**
- unsigned char **dirty**
- struct {
 - struct {
 - unsigned int **byte_idx**
 - unsigned int **char_idx**
 - Ewl_Text_Context * tx**
 - } **current**
 - Ecore_DList * nodes**
- } **formatting**
- unsigned char **in_select**
- struct {
 - unsigned int **bytes**
 - unsigned int **chars**

} **length**

- struct {
 - int **x**
 - int **y**
 } **offset**
- unsigned char **selectable**
- **Ewl_Text_Trigger** * **selection**
- char * **text**
- void * **textblock**
- unsigned int **total_size**
- **Ecore_List** * **triggers**

7.117.2 Field Documentation

7.117.2.1 unsigned int **Ewl_Text::byte_idx**

The current byte index

7.117.2.2 unsigned int **Ewl_Text::bytes**

Number of bytes in the text

7.117.2.3 unsigned int **Ewl_Text::char_idx**

The current char index

7.117.2.4 unsigned int **Ewl_Text::chars**

Number of characters in the text

7.117.2.5 **Ewl_Container** **Ewl_Text::container**

Inherit from **Ewl_Container** (p. 618)

7.117.2.6 struct { ... } **Ewl_Text::current**

Current text info

7.117.2.7 unsigned int **Ewl_Text::cursor_position**

The cursor position

7.117.2.8 unsigned char **Ewl_Text::delete_count**

Number of deletes

7.117.2.9 unsigned char Ewl_Text::dirty

Does the text need layout?

7.117.2.10 struct { ... } Ewl_Text::formatting

Holds the formatting information

7.117.2.11 unsigned char Ewl_Text::in_select

Are we in select mode?

7.117.2.12 struct { ... } Ewl_Text::length

Different text lengths

7.117.2.13 Ecore_DList* Ewl_Text::nodes

The formatting nodes

7.117.2.14 struct { ... } Ewl_Text::offset

Layout offset values

7.117.2.15 unsigned char Ewl_Text::selectable

Is the text selectable?

7.117.2.16 Ewl_Text_Trigger* Ewl_Text::selection

The current selection

7.117.2.17 char* Ewl_Text::text

The text itself

7.117.2.18 void* Ewl_Text::textblock

The Evas_Object_Textblock

7.117.2.19 unsigned int Ewl_Text::total_size

The total size we've alloc'd for text

7.117.2.20 Ecore_List* Ewl_Text::triggers

The list of triggers

7.117.2.21 `Ewl_Text_Context* Ewl_Text::tx`

The current formatting context

7.117.2.22 `int Ewl_Text::x`

X offset for layout

7.117.2.23 `int Ewl_Text::y`

Y offset for layout

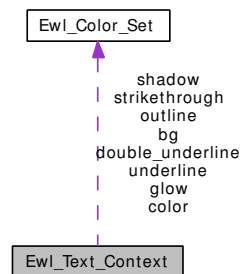
The documentation for this struct was generated from the following file:

- `lib/ewl_text.h`

7.118 Ewl_Text_Context Struct Reference

```
#include <ewl_text.h>
```

Collaboration diagram for Ewl_Text_Context:



7.118.1 Detailed Description

Stores context information for the different formatting nodes.

Data Fields

- unsigned int **align**
- **Ewl_Color_Set** color
- char * **font**
- char * **font_source**
- const char * **format**
- unsigned int **ref_count**
- char **size**
- struct {
 - Ewl_Color_Set** bg
 - Ewl_Color_Set** double_underline
 - Ewl_Color_Set** glow
 - Ewl_Color_Set** outline
 - Ewl_Color_Set** shadow
 - Ewl_Color_Set** strikethrough
 - Ewl_Color_Set** underline
 } **style_colors**
- unsigned int **styles**
- **Ewl_Text_Wrap** wrap

7.118.2 Field Documentation

7.118.2.1 unsigned int Ewl_Text_Context::align

Text alignment

7.118.2.2 Ewl_Color_Set Ewl_Text_Context::bg

Background colour

7.118.2.3 Ewl_Color_Set Ewl_Text_Context::color

Font colour

7.118.2.4 Ewl_Color_Set Ewl_Text_Context::double_underline

Double underline colour

7.118.2.5 char* Ewl_Text_Context::font

Font name

7.118.2.6 char* Ewl_Text_Context::font_source

The font source

7.118.2.7 const char* Ewl_Text_Context::format

This context's format string

7.118.2.8 Ewl_Color_Set Ewl_Text_Context::glow

Glow colour

7.118.2.9 Ewl_Color_Set Ewl_Text_Context::outline

Outline colour

7.118.2.10 unsigned int Ewl_Text_Context::ref_count

Number of references to this context

7.118.2.11 Ewl_Color_Set Ewl_Text_Context::shadow

Shadow colour

7.118.2.12 char Ewl_Text_Context::size

Font size

7.118.2.13 Ewl_Color_Set Ewl_Text_Context::strikethrough

Strikethrough colour

7.118.2.14 struct { ... } Ewl_Text_Context::style_colors

Colour information

7.118.2.15 unsigned int Ewl_Text_Context::styles

Styles set in this node

7.118.2.16 Ewl_Color_Set Ewl_Text_Context::underline

Underline colour

7.118.2.17 Ewl_Text_Wrap Ewl_Text_Context::wrap

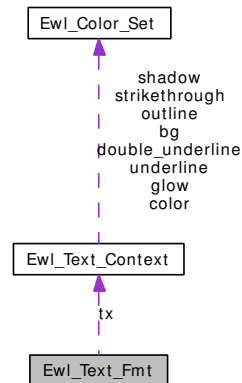
Text wrap setting

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

- lib/ewl_text.h

7.119 Ewl_Text_Fmt Struct Reference

Collaboration diagram for Ewl_Text_Fmt:



7.119.1 Detailed Description

Text formatting information.

Data Fields

- unsigned int `byte_len`
- unsigned int `char_len`
- **Ewl_Text_Context** * `tx`

7.119.2 Field Documentation

7.119.2.1 unsigned int **Ewl_Text_Fmt::byte_len**

7.119.2.2 unsigned int **Ewl_Text_Fmt::char_len**

7.119.2.3 **Ewl_Text_Context*** **Ewl_Text_Fmt::tx**

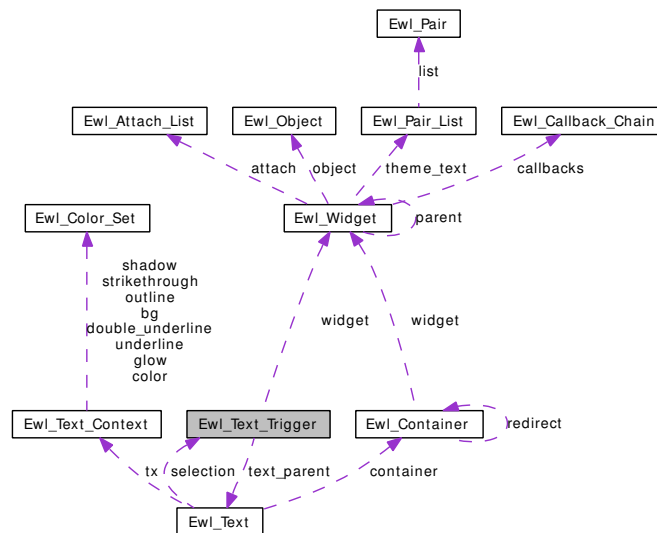
The documentation for this struct was generated from the following file:

- `lib/ewl_text.c`

7.120 Ewl_Text_Trigger Struct Reference

```
#include <ewl_text.h>
```

Collaboration diagram for Ewl_Text_Trigger:



7.120.1 Detailed Description

Inherits from `Ewl_Widget` (p. 785) and extends to provide a trigger for the text widget.

Data Fields

- `Ecore_List * areas`
- unsigned int `char_base`
- unsigned int `char_len`
- unsigned int `char_pos`
- `Ewl_Text * text_parent`
- `Ewl_Text_Trigger_Type type`
- `Ewl_Widget widget`

7.120.2 Field Documentation

7.120.2.1 `Ecore_List* Ewl_Text_Trigger::areas`

The list of objects for the trigger

7.120.2.2 unsigned int `Ewl_Text_Trigger::char_base`

Used for the selection. Start position

7.120.2.3 unsigned int Ewl_Text_Trigger::char_len

Trigger length

7.120.2.4 unsigned int Ewl_Text_Trigger::char_pos

Trigger start position

7.120.2.5 Ewl_Text* Ewl_Text_Trigger::text_parent

The parent text area

7.120.2.6 Ewl_Text_Trigger_Type Ewl_Text_Trigger::type

Trigger type

7.120.2.7 Ewl_Widget Ewl_Text_Trigger::widget

Inherit from widget

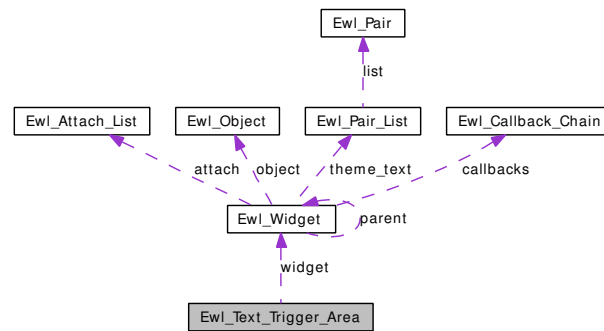
The documentation for this struct was generated from the following file:

- lib/**ewl_text.h**

7.121 Ewl_Text_Trigger_Area Struct Reference

```
#include <ewl_text.h>
```

Collaboration diagram for Ewl_Text_Trigger_Area:



7.121.1 Detailed Description

Inherits from **Ewl_Widget** (p. 785) and extends to provide a trigger area.

Data Fields

- unsigned int **deleted**
- **Ewl_Widget** widget

7.121.2 Field Documentation

7.121.2.1 unsigned int Ewl_Text_Trigger_Area::deleted

Is this area deleted

7.121.2.2 Ewl_Widget Ewl_Text_Trigger_Area::widget

Inherits from **Ewl_Widget** (p. 785)

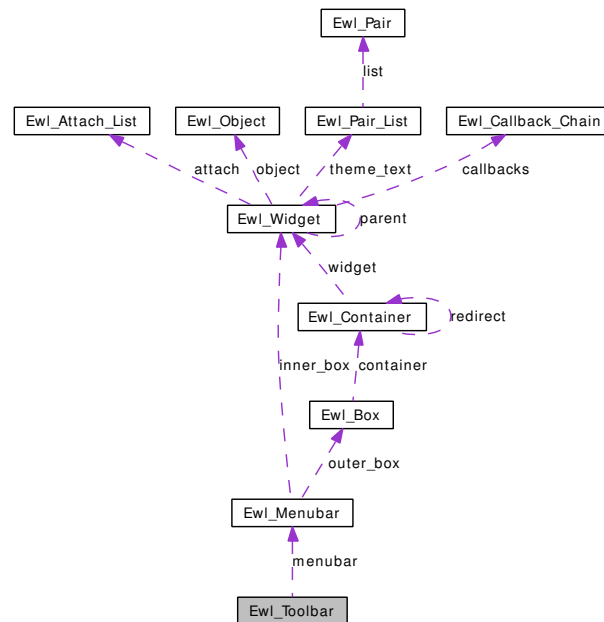
The documentation for this struct was generated from the following file:

- lib/ewl_text.h

7.122 Ewl_Toolbar Struct Reference

```
#include <ewl_toolbar.h>
```

Collaboration diagram for Ewl_Toolbar:



7.122.1 Detailed Description

Inherits from **Ewl_Menubar** (p. 701) and extends to provide a toolbar layout.

Data Fields

- **Ewl_Icon_Part** hidden
- **Ewl_Menubar** menubar

7.122.2 Field Documentation

7.122.2.1 Ewl_Icon_Part Ewl_Toolbar::hidden

which part of icons should be hidden

7.122.2.2 Ewl_Menubar Ewl_Toolbar::menubar

Inherit from **Ewl_Menubar** (p. 701)

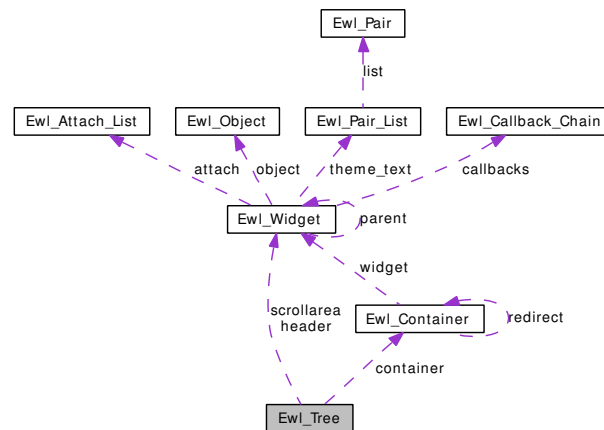
The documentation for this struct was generated from the following file:

- lib/ewl_toolbar.h

7.123 Ewl_Tree Struct Reference

```
#include <ewl_tree.h>
```

Collaboration diagram for Ewl_Tree:



7.123.1 Detailed Description

Inherits from **Ewl_Container** (p. 618) and extends to provide a tree layout.

Data Fields

- **Ewl_Container** container
- unsigned short **expands_visible**
- **Ewl_Widget * header**
- unsigned short **headers_visible**
- **Ewl_Selection_Mode** mode
- unsigned short **ncols**
- **Ewl_Widget * scrollarea**
- **Ecore_List * selected**

7.123.2 Field Documentation

7.123.2.1 Ewl_Container Ewl_Tree::container

Inherit from **Ewl_Container** (p. 618)

7.123.2.2 unsigned short Ewl_Tree::expands_visible

Are the rows expandable

7.123.2.3 Ewl_Widget* Ewl_Tree::header

Array of widgets in the header

7.123.2.4 unsigned short Ewl_Tree::headers_visible

Are the headers visible

7.123.2.5 Ewl_Selection_Mode Ewl_Tree::mode

Mode for selecting rows

7.123.2.6 unsigned short Ewl_Tree::ncols

Number of columns in tree

7.123.2.7 Ewl_Widget* Ewl_Tree::scrollarea

Scrollable area of rows

7.123.2.8 Ecore_List* Ewl_Tree::selected

The currently selected rows

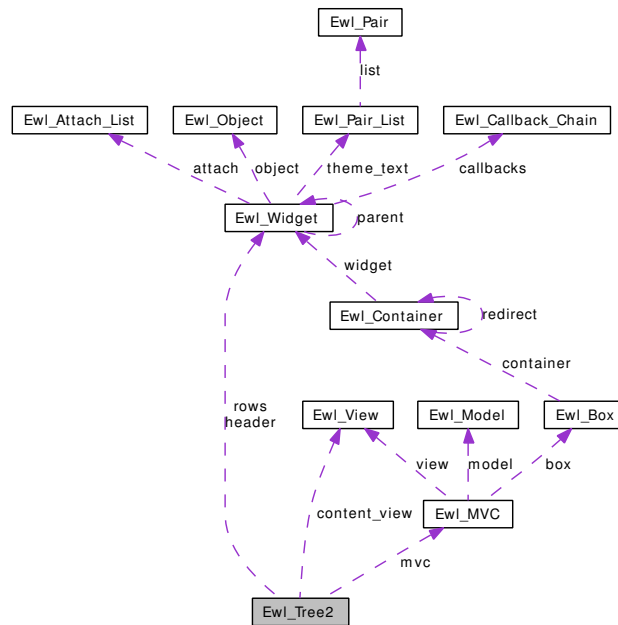
The documentation for this struct was generated from the following file:

- lib/**ewl_tree.h**

7.124 Ewl_Tree2 Struct Reference

```
#include <ewl_tree2.h>
```

Collaboration diagram for Ewl_Tree2:



7.124.1 Detailed Description

Inherits from `Ewl_MVC` (p. 705) and extends to provide a tree widget.

Data Fields

- unsigned int **columns**
- `Ewl_View *` **content_view**
- `Ecore_Hash *` **expansions**
- unsigned char **fixed**:1
- `Ewl_Widget *` **header**
- unsigned char **headers_visible**:1
- `Ewl_MVC` **mvc**
- int * **rowcache**
- `Ewl_Widget *` **rows**
- struct {
 - int **column**
 - `Ewl_Sort_Direction` **direction**
 } **sort**
- `Ewl_Tree_Selection_Type` **type**

7.124.2 Field Documentation

7.124.2.1 `int Ewl_Tree2::column`

The sort column

7.124.2.2 `unsigned int Ewl_Tree2::columns`

Number of columns in the tree

7.124.2.3 `Ewl_View* Ewl_Tree2::content_view`

View for the content widget

7.124.2.4 `Ewl_Sort_Direction Ewl_Tree2::direction`

Sort direction

7.124.2.5 `Ecore_Hash* Ewl_Tree2::expansions`

Hash of expanded rows

7.124.2.6 `unsigned char Ewl_Tree2::fixed`

Rows are fixed height

7.124.2.7 `Ewl_Widget* Ewl_Tree2::header`

The tree header

7.124.2.8 `unsigned char Ewl_Tree2::headers_visible`

Are the headers visible?

7.124.2.9 `Ewl_MVC Ewl_Tree2::mvc`

Inherit from `ewl_mvc`.

7.124.2.10 `int* Ewl_Tree2::rowcache`

Cache of row sizes

7.124.2.11 `Ewl_Widget* Ewl_Tree2::rows`

The rows of the tree

7.124.2.12 struct { ... } Ewl_Tree2::sort

The sort information

7.124.2.13 Ewl_Tree_Selection_Type Ewl_Tree2::type

The selection type of the tree

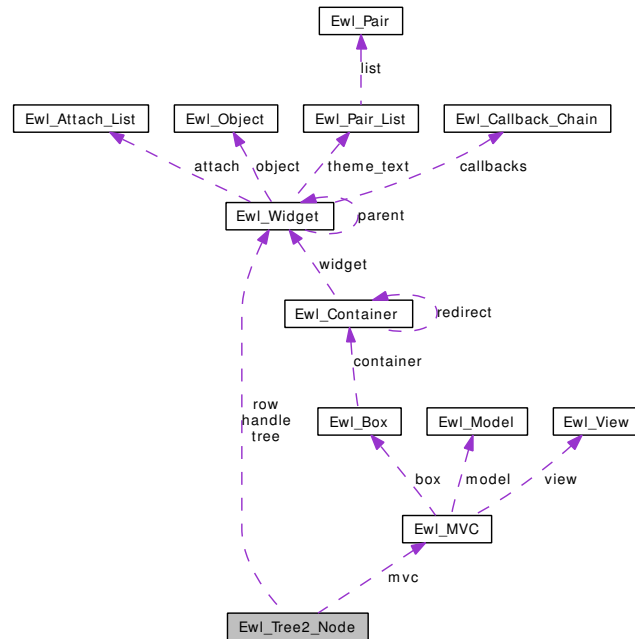
The documentation for this struct was generated from the following file:

- lib/ewl_tree2.h

7.125 Ewl_Tree2_Node Struct Reference

```
#include <ewl_tree2.h>
```

Collaboration diagram for Ewl_Tree2_Node:



7.125.1 Detailed Description

Inherits from `Ewl_Container` (p. 618) and extends to hold information on a row of the tree.

Data Fields

- unsigned char `built__children`:1
- `Ewl_Tree_Node_Flags` expanded
- `Ewl_Widget *` handle
- `Ewl_MVC` mvc
- `Ewl_Widget *` row
- unsigned int `row__num`
- `Ewl_Widget *` tree

7.125.2 Field Documentation

7.125.2.1 unsigned char Ewl_Tree2_Node::built__children

Have we generated child nodes

7.125.2.2 Ewl_Tree_Node_Flags Ewl_Tree2_Node::expanded

7.125.2.3 Ewl_Widget* Ewl_Tree2_Node::handle

the expansion handle

7.125.2.4 Ewl_MVC Ewl_Tree2_Node::mvc

7.125.2.5 Ewl_Widget* Ewl_Tree2_Node::row

The row this node is for

7.125.2.6 unsigned int Ewl_Tree2_Node::row_num

The row number of this row

7.125.2.7 Ewl_Widget* Ewl_Tree2_Node::tree

The parent tree

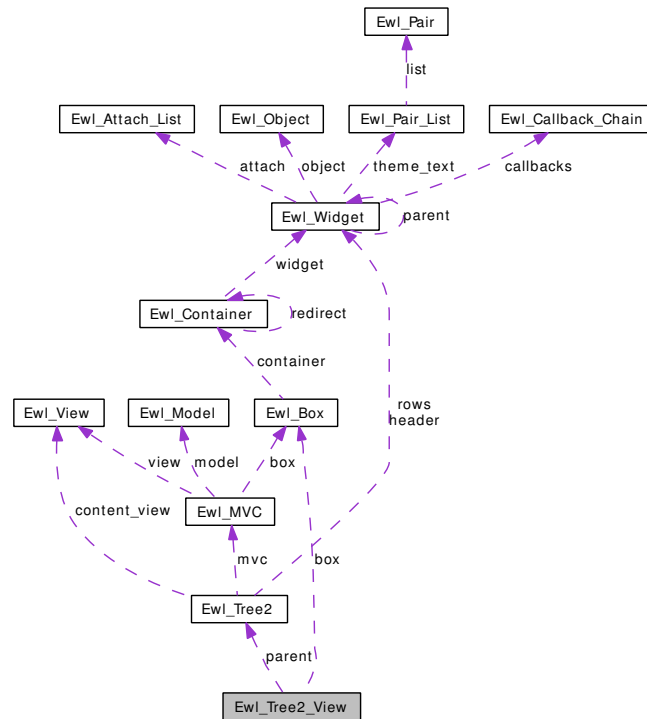
The documentation for this struct was generated from the following file:

- lib/ewl_tree2.h

7.126 Ewl_Tree2_View Struct Reference

```
#include <ewl_tree2_view.h>
```

Collaboration diagram for Ewl_Tree2_View:



7.126.1 Detailed Description

Inherits from **Ewl_Box** (p. 586) and provides the base information for a tree view.

Data Fields

- **Ewl_Box** box
- **Ewl_Tree2 *** parent

7.126.2 Field Documentation

7.126.2.1 Ewl_Box Ewl_Tree2_View::box

Inherit from **Ewl_Box** (p. 586)

7.126.2.2 Ewl_Tree2* Ewl_Tree2_View::parent

The tree parent

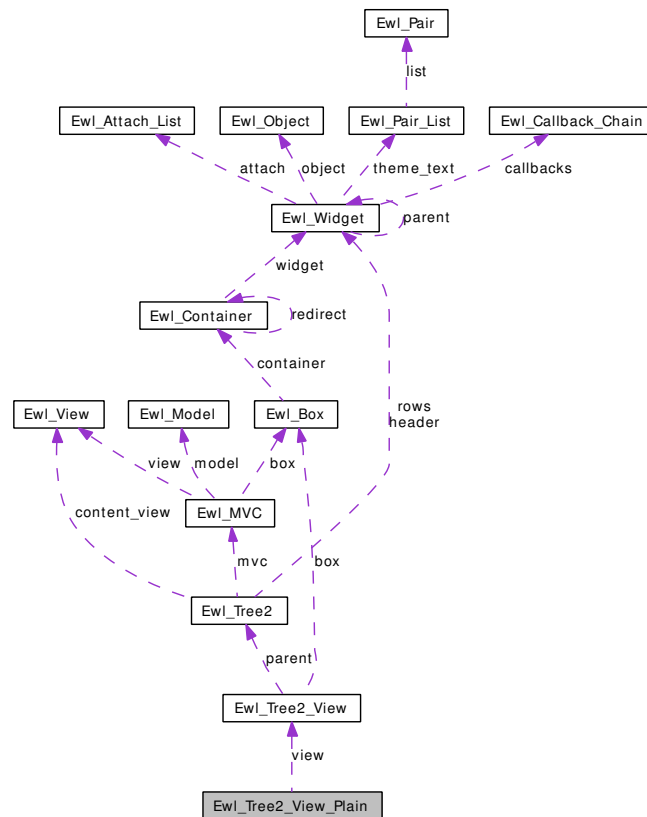
The documentation for this struct was generated from the following file:

- lib/ewl_tree2_view.h

7.127 Ewl_Tree2_View_Plain Struct Reference

```
#include <ewl_tree2_view_plain.h>
```

Collaboration diagram for Ewl_Tree2_View_Plain:



7.127.1 Detailed Description

Inherits from **Ewl_Tree2_View** (p. 776) and provides a plain tree layout.

Data Fields

- **Ewl_Tree2_View** view

7.127.2 Field Documentation

7.127.2.1 Ewl_Tree2_View Ewl_Tree2_View_Plain::view

Inherit from **Ewl_Tree2_View** (p. 776)

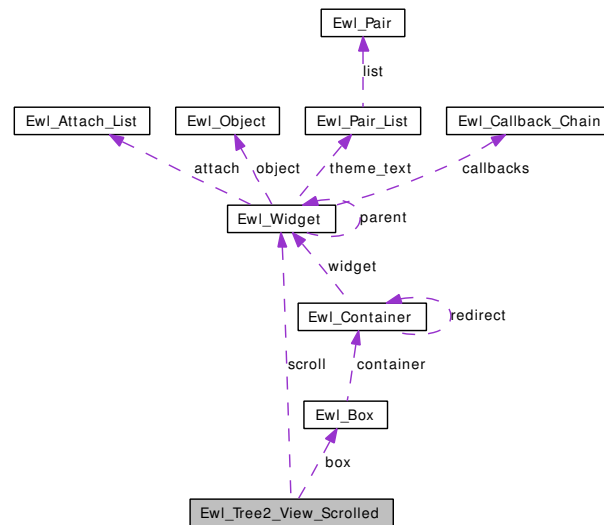
The documentation for this struct was generated from the following file:

- lib/ewl_tree2_view_plain.h

7.128 Ewl_Tree2_View_Scrolled Struct Reference

```
#include <ewl_tree2_view_scrolled.h>
```

Collaboration diagram for Ewl_Tree2_View_Scrolled:



7.128.1 Detailed Description

Inherits from **Ewl_Box** (p. 586) and provides a scrolled tree layout.

Data Fields

- **Ewl_Box** `box`
- **Ewl_Widget** * `scroll`
- unsigned char `scroll_header`:1
- unsigned char `scroll_visible`:1

7.128.2 Field Documentation

7.128.2.1 Ewl_Box Ewl_Tree2_View_Scrolled::box

Inherit from **Ewl_Box** (p. 586)

7.128.2.2 Ewl_Widget* Ewl_Tree2_View_Scrolled::scroll

The scroll region

7.128.2.3 unsigned char Ewl_Tree2_View_Scrolled::scroll_header

Do we scroll headers

7.128.2.4 unsigned char Ewl_Tree2_View_Scrolled::scroll_visible

Do we we show the headers

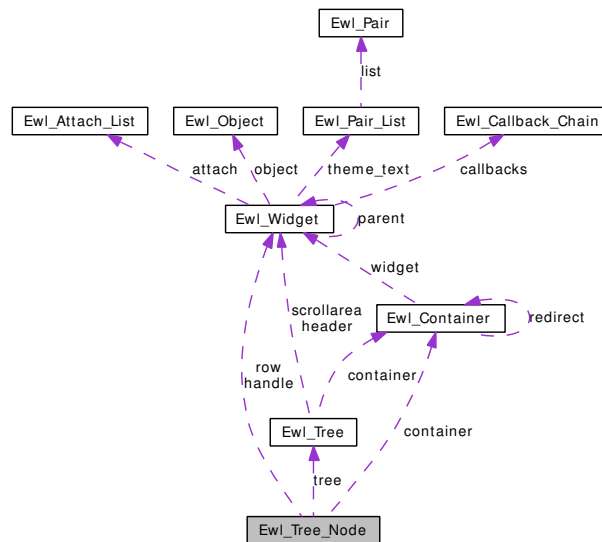
The documentation for this struct was generated from the following file:

- lib/ewl_tree2_view_scrolled.h

7.129 Ewl_Tree_Node Struct Reference

```
#include <ewl_tree.h>
```

Collaboration diagram for Ewl_Tree_Node:



7.129.1 Detailed Description

Inherits from **Ewl Container** (p.618) and extends to hold information on a row of the tree.

Data Fields

- **Ewl_Container** container
- **Ewl_Tree_Node_Flags** expanded
- **Ewl_Widget** * handle
- **Ewl_Widget** * row
- **Ewl_Tree** * tree

7.129.2 Field Documentation

7.129.2.1 Ewl_Container Ewl_Tree_Node::container

Inherit from **Ewl Container** (p. 618)

7.129.2.2 Ewl Tree Node Flags Ewl Tree Node::expanded

Indicator of expansion state

7.129.2.3 Ewl Widget* Ewl Tree Node::handle

The child row

7.129.2.4 Ewl_Widget* Ewl_Tree_Node::row

The child row

7.129.2.5 Ewl_Tree* Ewl_Tree_Node::tree

Pointer to the tree this is inside

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

- lib/**ewl_tree.h**

7.130 Ewl_Unit_Test Struct Reference

```
#include <Ewl_Test.h>
```

7.130.1 Detailed Description

Contains the information needed to run a unit test.

Data Fields

- char * **failure_reason**
- int(* **func**)(char *buf, int len)
- const char * **name**
- short **status**

7.130.2 Field Documentation

7.130.2.1 char* Ewl_Unit_Test::failure_reason

Test failure reason

7.130.2.2 int(* Ewl_Unit_Test::func)(char *buf, int len)

The test function

7.130.2.3 const char* Ewl_Unit_Test::name

The test name

7.130.2.4 short Ewl_Unit_Test::status

Test status

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

- bin/Ewl_Test.h

7.131 Ewl_View Struct Reference

```
#include <ewl_view.h>
```

7.131.1 Detailed Description

The view function pointers.

Data Fields

- **Ewl_View_Expansion_View_Fetch** expansion
- **Ewl_View_Widget_Fetch** fetch
- **Ewl_View_Header_Fetch** header_fetch

7.131.2 Field Documentation

7.131.2.1 Ewl_View_Expansion_View_Fetch Ewl_View::expansion

Get a view for displaying expansion data

7.131.2.2 Ewl_View_Widget_Fetch Ewl_View::fetch

Get a new widget for display

7.131.2.3 Ewl_View_Header_Fetch Ewl_View::header_fetch

Get the header for the given column

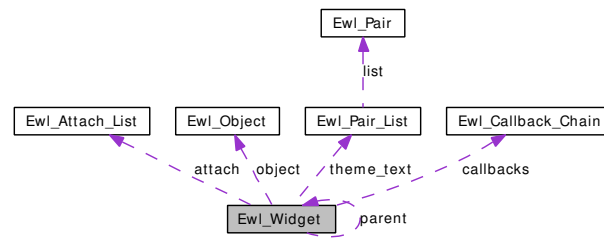
The documentation for this struct was generated from the following file:

- **lib/ewl_view.h**

7.132 Ewl_Widget Struct Reference

```
#include <ewl_widget.h>
```

Collaboration diagram for Ewl_Widget:



7.132.1 Detailed Description

Inherits from **Ewl_Object** (p. 709) and extends to provide appearance, parent, and callback capabilities.

Data Fields

- `const char *` **appearance**
- `Ewl_Attach_List *` **attach**
- `Ewl_Callback_Chain` **callbacks** [EWL_CALLBACK_MAX+1]
- `void *` **fx_clip_box**
- `const char *` **inheritance**
- `int` **layer**
- `Ewl_Object` **object**
- `Ewl_Widget *` **parent**
- `void *` **smart_object**
- `Ecore_Hash *` **theme**
- `const char *` **theme_group**
- `void *` **theme_object**
- `const char *` **theme_path**
- `const char *` **theme_state**
- `Ewl_Pair_List` **theme_text**
- `int` **toplayered**

7.132.2 Field Documentation

7.132.2.1 `const char* Ewl_Widget::appearance`

Key to lookup appearance in theme

7.132.2.2 `Ewl_Attach_List* Ewl_Widget::attach`

List of attachments on the widget

7.132.2.3 Ewl_Callback_Chain Ewl_Widget::callbacks[EWL_CALLBACK_MAX+1]

Callback chain array

7.132.2.4 void* Ewl_Widget::fx_clip_box

Clipping rectangle of widget

7.132.2.5 const char* Ewl_Widget::inheritance

Inheritance of path widget

7.132.2.6 int Ewl_Widget::layer

the layer relative to the parent

7.132.2.7 Ewl_Object Ewl_Widget::object

Inherit the base Object class

7.132.2.8 Ewl_Widget* Ewl_Widget::parent

The parent widget, actually a container

7.132.2.9 void* Ewl_Widget::smart_object

Smart Object for the layer stuff

7.132.2.10 Ecore_Hash* Ewl_Widget::theme

Overriding theme settings

7.132.2.11 const char* Ewl_Widget::theme_group

Group in theme to use

7.132.2.12 void* Ewl_Widget::theme_object

Appearance shown on canvas

7.132.2.13 const char* Ewl_Widget::theme_path

Path to the file for loading

7.132.2.14 `const char* Ewl_Widget::theme_state`

State of the appearance

7.132.2.15 `Ewl_Pair_List Ewl_Widget::theme_text`

Overriding text in theme

7.132.2.16 `int Ewl_Widget::toplayered`

Indicates if the widget should be on the top of the layer stack

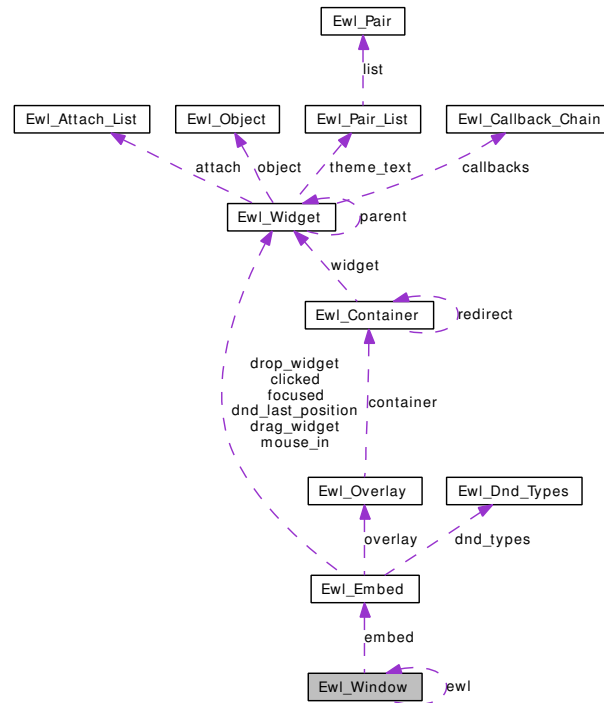
The documentation for this struct was generated from the following file:

- `lib/ewl_widget.h`

7.133 Ewl_Window Struct Reference

```
#include <ewl_window.h>
```

Collaboration diagram for Ewl_Window:



7.133.1 Detailed Description

Inherits from **Ewl_Embed** (p. 629) class to create it's own window and canvas for drawing, sizing and positioning.

Data Fields

- `char * classname`
- `Ewl_Embed embed`
- `Ewl_Window_Flags flags`
- `char * name`
- `Ewl_Window_Flags old_flags`
- `char * title`
- `union {`
 - `Ewl_Window * ewl`
 - `Ewl_Embed_Window * foreign`
- `} transient`
- `void * window`

7.133.2 Field Documentation

7.133.2.1 char* Ewl_Window::classname

Current class on the provided window

7.133.2.2 Ewl_Embed Ewl_Window::embed

Inherits from the **Ewl_Embed** (p. 629) class

7.133.2.3 Ewl_Window* Ewl_Window::ewl

7.133.2.4 Ewl_Window_Flags Ewl_Window::flags

Flags indicating window properties

7.133.2.5 Ewl_Embed_Window* Ewl_Window::foreign

7.133.2.6 char* Ewl_Window::name

Current name on the provided window

7.133.2.7 Ewl_Window_Flags Ewl_Window::old_flags

Flags used on our last state set call

7.133.2.8 char* Ewl_Window::title

The current title on the provided window

7.133.2.9 union { ... } Ewl_Window::transient

Window to be transient for

7.133.2.10 void* Ewl_Window::window

Provides a window for drawing

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

- lib/ewl_window.h

7.134 Freebox__Test Struct Reference

Data Fields

- **Ewl_Freebox__Comparator** compare
- `const char *` **name**
- `const Ewl_Freebox__Layout_Type` type

7.134.1 Field Documentation

7.134.1.1 `Ewl_Freebox__Comparator` `Freebox__Test::compare`

7.134.1.2 `const char*` `Freebox__Test::name`

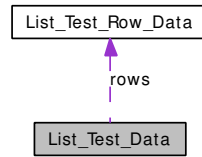
7.134.1.3 `const Ewl_Freebox__Layout_Type` `Freebox__Test::type`

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

- `bin/tests/freebox/ewl_freebox.c`

7.135 List_Test_Data Struct Reference

Collaboration diagram for List_Test_Data:



Data Fields

- unsigned int `count`
- `List_Test_Row_Data ** rows`

7.135.1 Field Documentation

7.135.1.1 unsigned int `List_Test_Data::count`

7.135.1.2 `List_Test_Row_Data** List_Test_Data::rows`

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

- `bin/tests/list/ewl_list.c`

7.136 List_Test_Row_Data Struct Reference

Data Fields

- char * **image**
- char * **text**

7.136.1 Field Documentation

7.136.1.1 char* List_Test_Row_Data::image

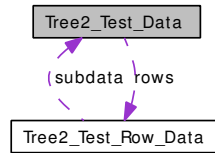
7.136.1.2 char* List_Test_Row_Data::text

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

- bin/tests/list/**ewl_list.c**

7.137 Tree2_Test_Data Struct Reference

Collaboration diagram for Tree2_Test_Data:



Data Fields

- unsigned int **count**
- Tree2_Test_Row_Data ** rows

7.137.1 Field Documentation

7.137.1.1 unsigned int Tree2_Test_Data::count

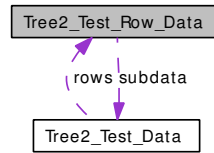
7.137.1.2 Tree2_Test_Row_Data** Tree2_Test_Data::rows

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

- bin/tests/tree2/ewl_tree2.c

7.138 Tree2_Test_Row_Data Struct Reference

Collaboration diagram for Tree2_Test_Row_Data:



Data Fields

- int **expandable**
- char * **image**
- Tree2_Test_Data * **subdata**
- char * **text**

7.138.1 Field Documentation

7.138.1.1 int Tree2_Test_Row_Data::expandable

7.138.1.2 char* Tree2_Test_Row_Data::image

7.138.1.3 Tree2_Test_Data* Tree2_Test_Row_Data::subdata

7.138.1.4 char* Tree2_Test_Row_Data::text

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

- bin/tests/tree2/ewl_tree2.c

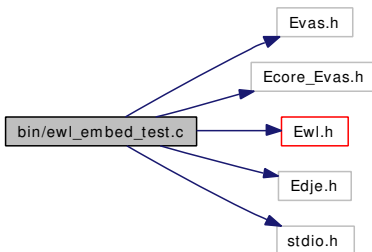
Chapter 8

Enlightened Widget Library File Documentation

8.1 bin/ewl_embed_test.c File Reference

```
#include <Evas.h>
#include <Ecore_Evas.h>
#include <Ewl.h>
#include <Edje.h>
#include <stdio.h>
```

Include dependency graph for ewl_embed_test.c:



Defines

- `#define __UNUSED__`

Functions

- `int main (int argc, char **argv)`

Variables

- `Evas_Object * edje`

- `Ecore_Evas * ee`
- `Ewl_Widget * entry`
- `Evas * evas`
- `Ewl_Widget * text`

8.1.1 Define Documentation

8.1.1.1 `#define __UNUSED__`

8.1.2 Function Documentation

8.1.2.1 `int main (int argc, char ** argv)`

8.1.3 Variable Documentation

8.1.3.1 `Evas_Object* edje`

8.1.3.2 `Ecore_Evas* ee`

8.1.3.3 `Ewl_Widget * entry`

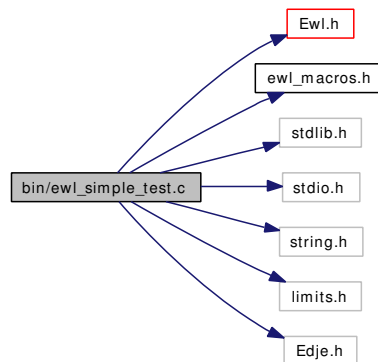
8.1.3.4 `Evas* evas`

8.1.3.5 `Ewl_Widget* text`

8.2 bin/ewl_simple_test.c File Reference

```
#include <Ewl.h>
#include <ewl_macros.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <limits.h>
#include <Edge.h>
```

Include dependency graph for ewl_simple_test.c:



Defines

- `#define __UNUSED__`

Functions

- `int main (int argc, char **argv)`

8.2.1 Define Documentation

8.2.1.1 `#define __UNUSED__`

8.2.2 Function Documentation

8.2.2.1 `int main (int argc, char ** argv)`

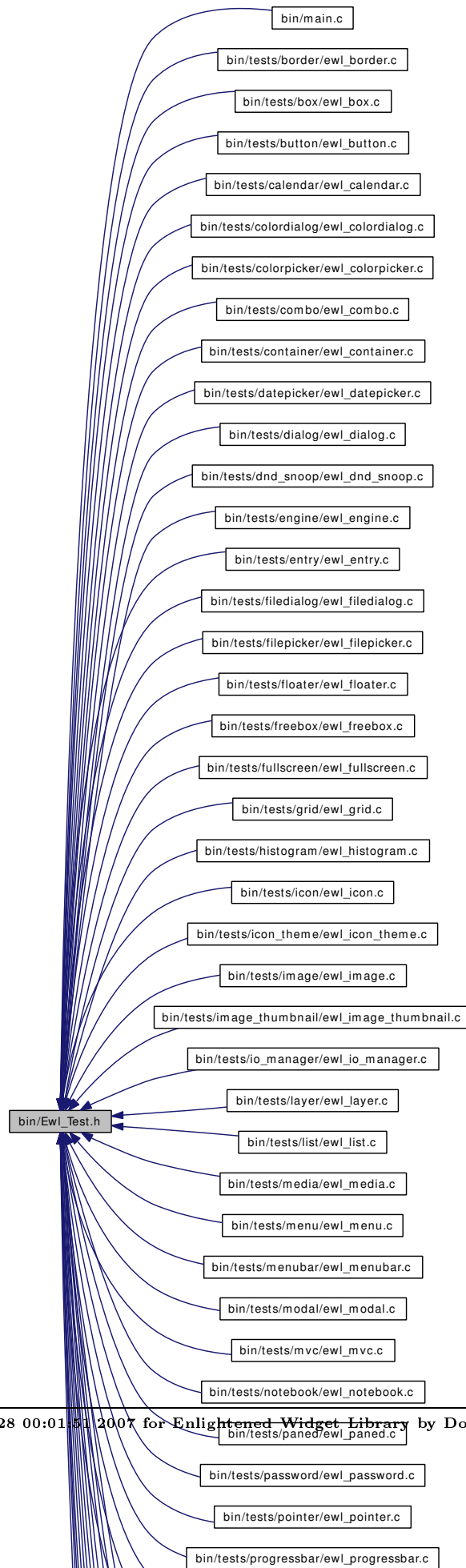
8.3 bin/Ewl_Test.h File Reference

```
#include <Ewl.h>
```

Include dependency graph for Ewl_Test.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Test**
Holds information on an Ewl Test set.
- struct **Ewl_Unit_Test**
Contains the information needed to run a unit test.

Typedefs

- typedef **Ewl_Test** **Ewl_Test**
- typedef enum **Ewl_Test_Type** **Ewl_Test_Type**
- typedef **Ewl_Unit_Test** **Ewl_Unit_Test**

Enumerations

- enum **Ewl_Test_Type** {
 EWL_TEST_TYPE_SIMPLE, **EWL_TEST_TYPE_ADVANCED**, **EWL_TEST_TYPE_CONTAINER**, **EWL_TEST_TYPE_MISC**,
 EWL_TEST_TYPE_UNIT }

8.3.1 Typedef Documentation

8.3.1.1 typedef struct Ewl_Test Ewl_Test

Provides information on an Ewl Test set

8.3.1.2 typedef enum Ewl_Test_Type Ewl_Test_Type

8.3.1.3 typedef struct Ewl_Unit_Test Ewl_Unit_Test

Provides information on a unit test

8.3.2 Enumeration Type Documentation

8.3.2.1 enum Ewl_Test_Type

The different types of tests available

Enumerator:

EWL_TEST_TYPE_SIMPLE
EWL_TEST_TYPE_ADVANCED
EWL_TEST_TYPE_CONTAINER
EWL_TEST_TYPE_MISC
EWL_TEST_TYPE_UNIT

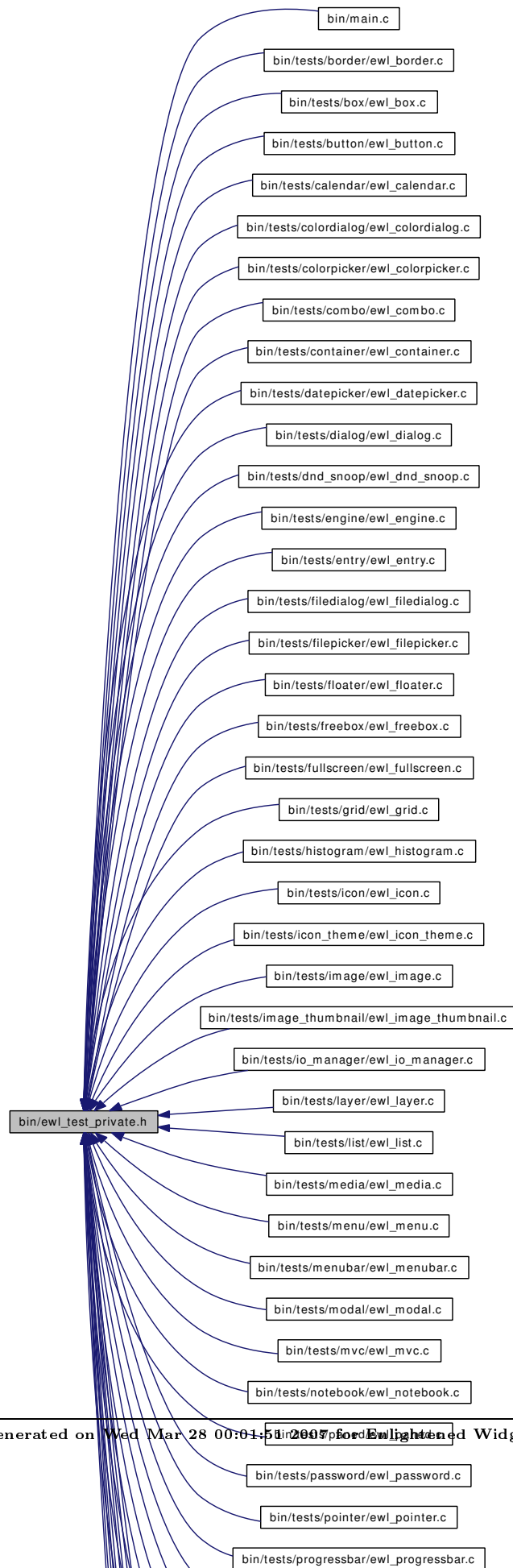
8.4 bin/ewl_test_private.h File Reference

```
#include <ewl-config.h>
```

Include dependency graph for ewl_test_private.h:



This graph shows which files directly or indirectly include this file:



Defines

- `#define __UNUSED__`

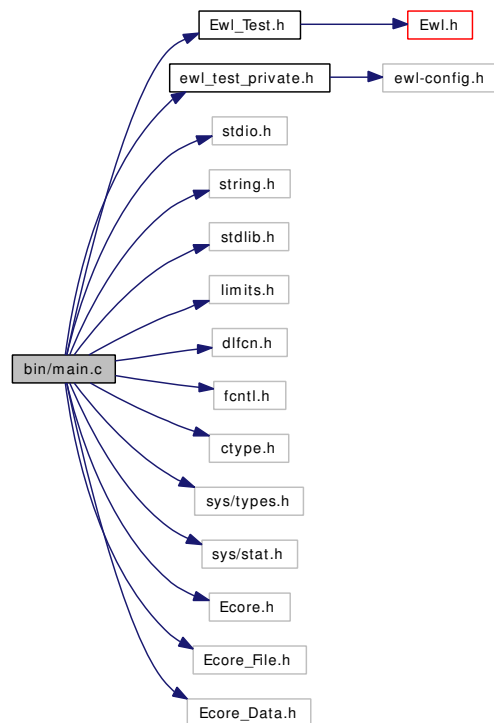
8.4.1 Define Documentation

8.4.1.1 `#define __UNUSED__`

8.5 bin/main.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <limits.h>
#include <dlfcn.h>
#include <fcntl.h>
#include <ctype.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <Ecore.h>
#include <Ecore_File.h>
#include <Ecore_Data.h>
```

Include dependency graph for main.c:



Defines

- `#define MAIN_HEIGHT 320`

- `#define MAIN__WIDTH 640`

Functions

- `int main (int argc, char **argv)`

8.5.1 Define Documentation

8.5.1.1 `#define MAIN__HEIGHT 320`

8.5.1.2 `#define MAIN__WIDTH 640`

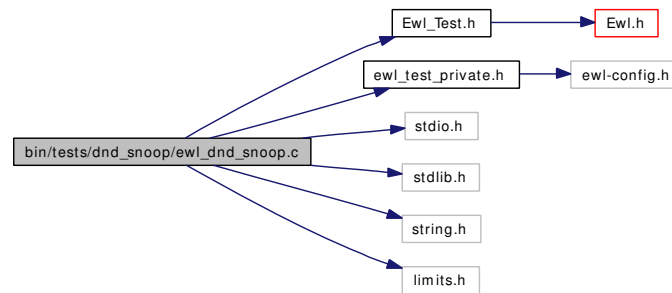
8.5.2 Function Documentation

8.5.2.1 `int main (int argc, char ** argv)`

8.6 bin/tests/dnd_snoop/ewl_dnd_snoop.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <limits.h>
```

Include dependency graph for ewl_dnd_snoop.c:



Functions

- void **test_info** (Ewl_Test *test)

Variables

- Ecore_Event_Handler * **ewl_dnd_client_message_handler** = NULL
- Ecore_Event_Handler * **ewl_dnd_drop_handler** = NULL
- Ecore_Event_Handler * **ewl_dnd_enter_handler** = NULL
- Ecore_Event_Handler * **ewl_dnd_finished_handler** = NULL
- Ecore_Event_Handler * **ewl_dnd_leave_handler** = NULL
- Ecore_Event_Handler * **ewl_dnd_position_handler** = NULL
- Ecore_Event_Handler * **ewl_dnd_selection_clear_handler** = NULL
- Ecore_Event_Handler * **ewl_dnd_selection_notify_handler** = NULL
- Ecore_Event_Handler * **ewl_dnd_selection_request_handler** = NULL
- Ecore_Event_Handler * **ewl_dnd_status_handler** = NULL

8.6.1 Function Documentation

8.6.1.1 void test_info (Ewl_Test * *test*)

8.6.2 Variable Documentation

8.6.2.1 Ecore_Event_Handler* ewl_dnd_client_message_handler = NULL

8.6.2.2 Ecore_Event_Handler* ewl_dnd_drop_handler = NULL

8.6.2.3 Ecore_Event_Handler* ewl_dnd_enter_handler = NULL

8.6.2.4 Ecore_Event_Handler* ewl_dnd_finished_handler = NULL

8.6.2.5 Ecore_Event_Handler* ewl_dnd_leave_handler = NULL

8.6.2.6 Ecore_Event_Handler* ewl_dnd_position_handler = NULL

8.6.2.7 Ecore_Event_Handler* ewl_dnd_selection_clear_handler = NULL

8.6.2.8 Ecore_Event_Handler* ewl_dnd_selection_notify_handler = NULL

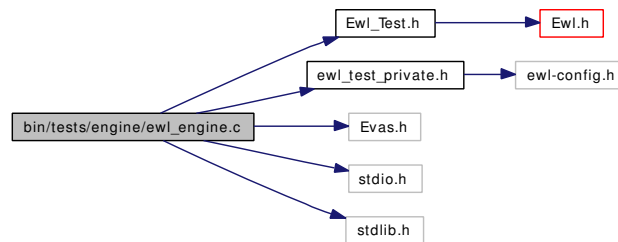
8.6.2.9 Ecore_Event_Handler* ewl_dnd_selection_request_handler = NULL

8.6.2.10 Ecore_Event_Handler* ewl_dnd_status_handler = NULL

8.7 bin/tests/engine/ewl_engine.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <Evas.h>
#include <stdio.h>
#include <stdlib.h>
```

Include dependency graph for ewl_engine.c:



Defines

- `#define WIN_NAME "buffer_engine_window"`

Functions

- `void test_info (Ewl_Test *test)`

8.7.1 Define Documentation

8.7.1.1 `#define WIN_NAME "buffer_engine_window"`

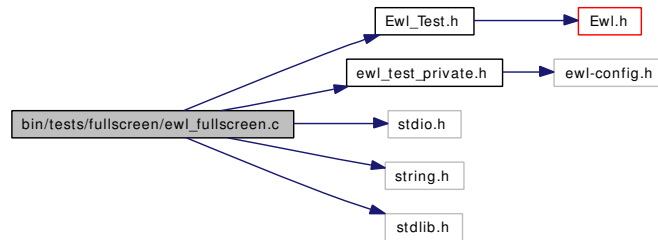
8.7.2 Function Documentation

8.7.2.1 `void test_info (Ewl_Test * test)`

8.8 bin/tests/fullscreen/ewl_fullscreen.c File Reference

```
#include "Ewl_Test.h"  
#include "ewl_test_private.h"  
#include <stdio.h>  
#include <string.h>  
#include <stdlib.h>
```

Include dependency graph for ewl_fullscreen.c:



Functions

- `void test_info (Ewl_Test *test)`

8.8.1 Function Documentation

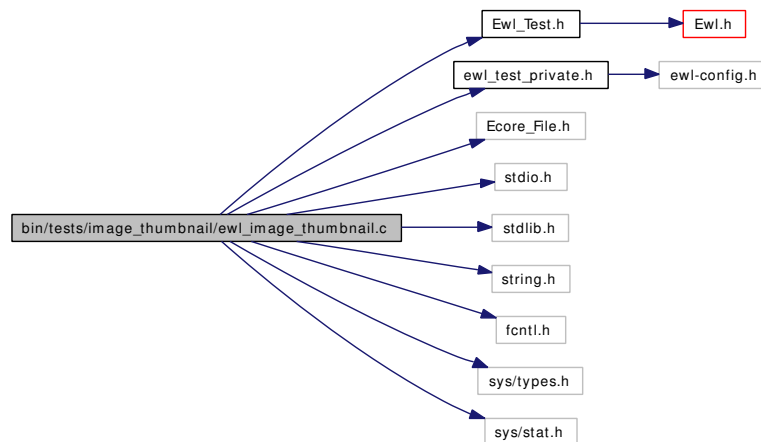
8.8.1.1 `void test_info (Ewl_Test * test)`

8.9 bin/tests/image_thumbnail/ewl_image_thumbnail.c

File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <Ecore_File.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
```

Include dependency graph for ewl_image_thumbnail.c:



Functions

- `void test_info (Ewl_Test *test)`

8.9.1 Function Documentation

8.9.1.1 `void test_info (Ewl_Test * test)`

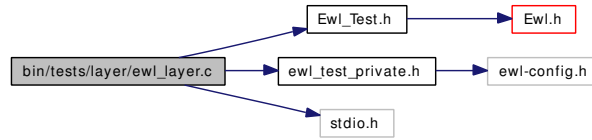
8.10 bin/tests/layer/ewl_layer.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_layer.c:



Functions

- `void test_info (Ewl_Test *test)`

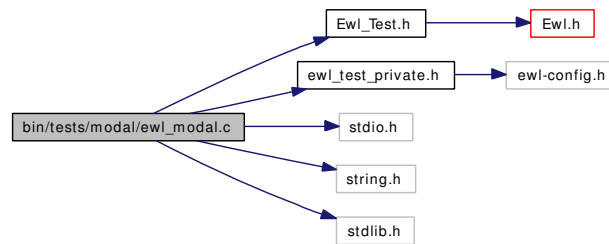
8.10.1 Function Documentation

8.10.1.1 `void test_info (Ewl_Test * test)`

8.11 bin/tests/modal/ewl_modal.c File Reference

```
#include "Ewl_Test.h"  
#include "ewl_test_private.h"  
#include <stdio.h>  
#include <string.h>  
#include <stdlib.h>
```

Include dependency graph for ewl_modal.c:



Functions

- `void test_info (Ewl_Test *test)`

8.11.1 Function Documentation

8.11.1.1 `void test_info (Ewl_Test * test)`

8.12 bin/tests/pointer/ewl_pointer.c File Reference

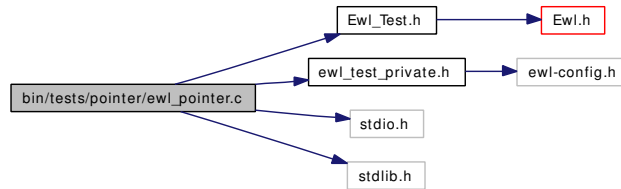
```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

Include dependency graph for ewl_pointer.c:



Functions

- `void test_info (Ewl_Test *test)`

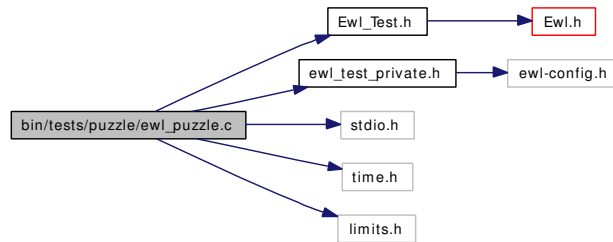
8.12.1 Function Documentation

8.12.1.1 `void test_info (Ewl_Test * test)`

8.13 bin/tests/puzzle/ewl_puzzle.c File Reference

```
#include "Ewl_Test.h"  
#include "ewl_test_private.h"  
#include <stdio.h>  
#include <time.h>  
#include <limits.h>
```

Include dependency graph for ewl_puzzle.c:



Functions

- `void test_info (Ewl_Test *test)`

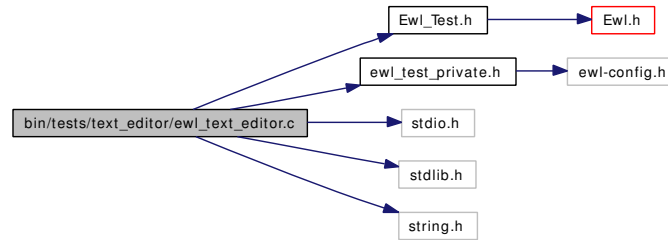
8.13.1 Function Documentation

8.13.1.1 `void test_info (Ewl_Test * test)`

8.14 bin/tests/text_editor/ewl_text_editor.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for ewl_text_editor.c:



Data Structures

- struct **Control**

Functions

- void **test_info** (Ewl_Test *test)

8.14.1 Function Documentation

8.14.1.1 void test_info (Ewl_Test * test)

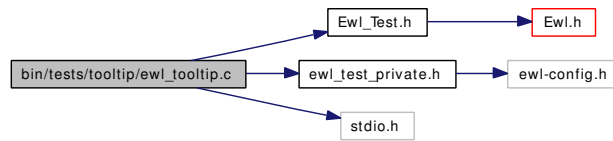
8.15 bin/tests/tooltip/ewl_tooltip.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_tooltip.c:



Functions

- `void test_info (Ewl_Test *test)`

8.15.1 Function Documentation

8.15.1.1 `void test_info (Ewl_Test * test)`

8.16 lib/Ewl.h File Reference

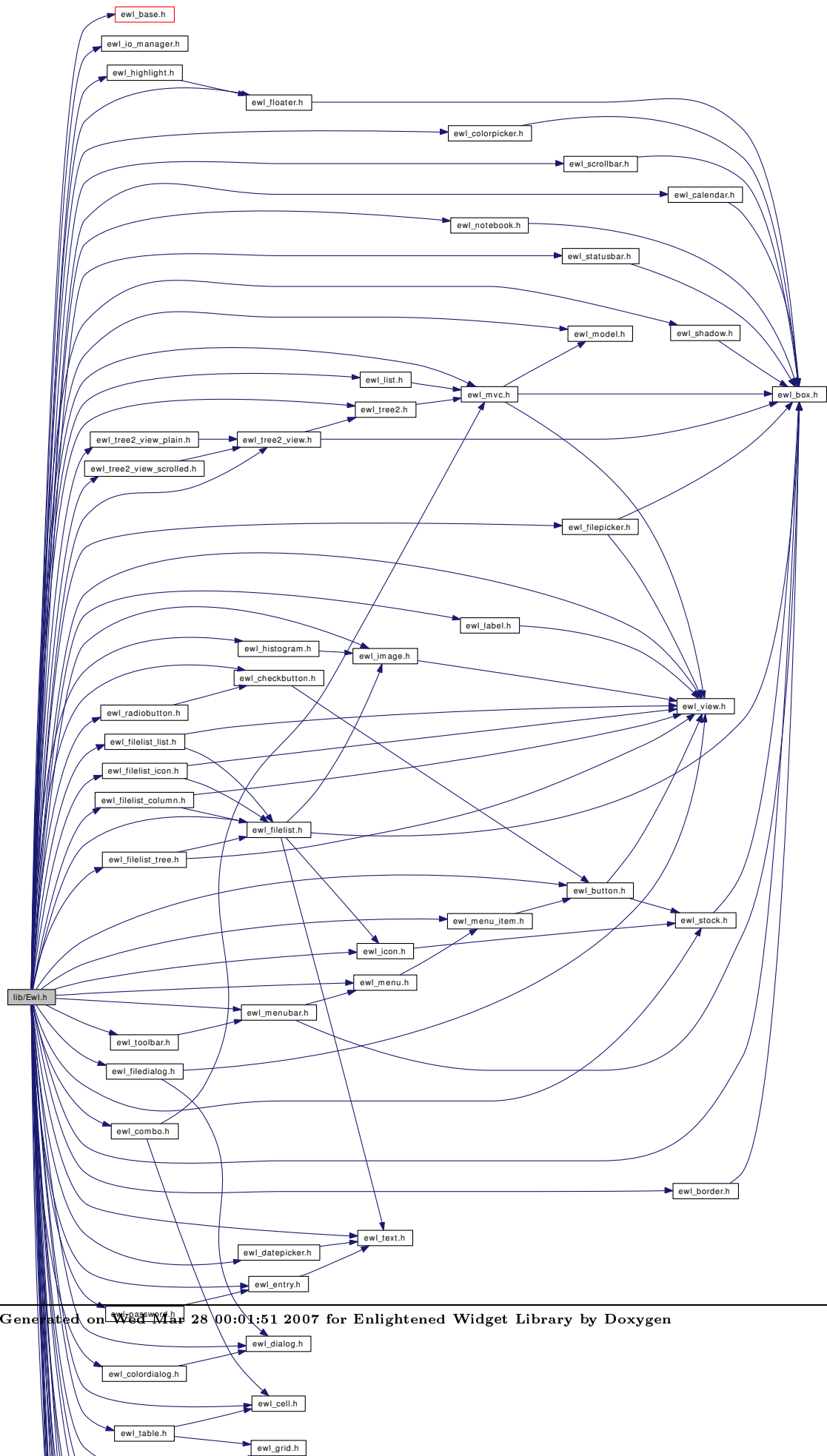
8.16.1 Detailed Description

The file that should be included by any project using EWL. Provides all the necessary headers and includes to work with EWL. It is discouraged to include each header file individually.

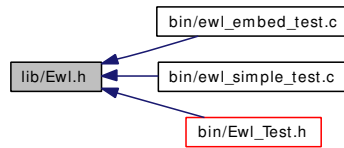
```
#include <ewl_base.h>
#include <ewl_io_manager.h>
#include <ewl_model.h>
#include <ewl_view.h>
#include <ewl_box.h>
#include <ewl_border.h>
#include <ewl_cell.h>
#include <ewl_row.h>
#include <ewl_grid.h>
#include <ewl_table.h>
#include <ewl_label.h>
#include <ewl_stock.h>
#include <ewl_button.h>
#include <ewl_floater.h>
#include <ewl_dialog.h>
#include <ewl_popup.h>
#include <ewl_text.h>
#include <ewl_entry.h>
#include <ewl_colorpicker.h>
#include <ewl_colordialog.h>
#include <ewl_password.h>
#include <ewl_range.h>
#include <ewl_seeker.h>
#include <ewl_scrollbar.h>
#include <ewl_spacer.h>
#include <ewl_spinner.h>
#include <ewl_image.h>
#include <ewl_highlight.h>
#include <ewl_histogram.h>
#include <ewl_spectrum.h>
#include <ewl_menu_item.h>
#include <ewl_menu.h>
```

```
#include <ewl_menubar.h>
#include <ewl_context_menu.h>
#include <ewl_toolbar.h>
#include <ewl_check.h>
#include <ewl_checkbutton.h>
#include <ewl_radiobutton.h>
#include <ewl_separator.h>
#include <ewl_calendar.h>
#include <ewl_datepicker.h>
#include <ewl_icon.h>
#include <ewl_icon_theme.h>
#include <ewl_notebook.h>
#include <ewl_progressbar.h>
#include <ewl_paned.h>
#include <ewl_scrollpane.h>
#include <ewl_statusbar.h>
#include <ewl_shadow.h>
#include <ewl_mvc.h>
#include <ewl_combo.h>
#include <ewl_tree.h>
#include <ewl_tree2.h>
#include <ewl_tree2_view.h>
#include <ewl_tree2_view_plain.h>
#include <ewl_tree2_view_scrolled.h>
#include <ewl_list.h>
#include <ewl_freebox.h>
#include <ewl_filelist.h>
#include <ewl_filelist_list.h>
#include <ewl_filelist_icon.h>
#include <ewl_filelist_column.h>
#include <ewl_filelist_tree.h>
#include <ewl_filepicker.h>
#include <ewl_filedialog.h>
#include <ewl_media.h>
```

Include dependency graph for Ewl.h:



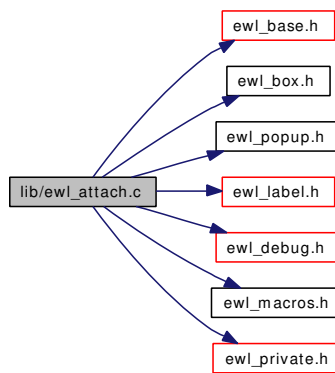
This graph shows which files directly or indirectly include this file:



8.17 lib/ewl_attach.c File Reference

```
#include "ewl_base.h"
#include "ewl_box.h"
#include "ewl_popup.h"
#include "ewl_label.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_attach.c:



Data Structures

- struct **Ewl_Attach_Tooltip**

*Inherits from **Ewl_Widget** (p. 785) and extends to provide information on a tooltip.*

Typedefs

- typedef **Ewl_Attach_Tooltip** **Ewl_Attach_Tooltip**

Functions

- void **ewl_attach_dnd_drag_set** (**Ewl_Widget** *w, **Ewl_Widget** *c, void *data, int size)
Attaches the DND data data to the widget w with the displayed cursor c.
- void * **ewl_attach_get** (**Ewl_Widget** *w, **Ewl_Attach_Type** t)
Get the attachment of type t from the widget w.
- void **ewl_attach_list_del** (**Ewl_Attach_List** *list, **Ewl_Attach_Type** type)
Deletes the given type type from the list list.

- void `ewl_attach_other_set` (`Ewl_Widget *w`, `Ewl_Attach_Type t`, void `*data`)
Attaches the data `data` to the widget `w` with the `attache` type of `t`.
- void `ewl_attach_text_set` (`Ewl_Widget *w`, `Ewl_Attach_Type t`, const char `*data`)
Attaches the text `data` to the widget `w`.
- void `ewl_attach_widget_set` (`Ewl_Widget *w`, `Ewl_Attach_Type t`, `Ewl_Widget *data`)
Attaches a widget `data` to the widget `w`.

8.17.1 Typedef Documentation

8.17.1.1 typedef struct Ewl_Attach_Tooltip Ewl_Attach_Tooltip

`Ewl_Attach_Tooltip` (p. 582)

8.17.2 Function Documentation

8.17.2.1 void `ewl_attach_dnd_drag_set` (`Ewl_Widget * w`, `Ewl_Widget * c`, void `* data`, int `size`)

Attaches the DND data `data` to the widget `w` with the displayed cursor `c`.

Parameters:

- `w`,:* The widget to attach the dnd data too
- `c`,:* The cursor to display during drag.
- `data`,:* The data to transfer on drop.
- `size`,:* The amount of data send

Returns:

Returns no value

8.18 lib/ewl_attach.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Attach**
Contains information about widget attachments.
- struct **Ewl_Attach_Dnd**
DND data associated with the widget.

Defines

- #define **EWL_ATTACH**(attach) ((Ewl_Attach *) attach)
- #define **ewl_attach_color_get**(w) ewl_attach_get(EWL_WIDGET(w), EWL_ATTACH_TYPE_COLOR)
- #define **ewl_attach_color_set**(w, data) ewl_attach_other_set(EWL_WIDGET(w), EWL_ATTACH_TYPE_COLOR, data)
- #define **EWL_ATTACH_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w)), EWL_ATTACH_TYPE)
- #define **ewl_attach_mouse_argb_cursor_set**(w, cursor)
- #define **ewl_attach_mouse_cursor_set**(w, data)
- #define **ewl_attach_name_get**(w) ewl_attach_get(EWL_WIDGET(w), EWL_ATTACH_TYPE_NAME)
- #define **ewl_attach_name_set**(w, data) ewl_attach_text_set(EWL_WIDGET(w), EWL_ATTACH_TYPE_NAME, data)
- #define **ewl_attach_tooltip_text_set**(w, data) ewl_attach_text_set(EWL_WIDGET(w), EWL_ATTACH_TYPE_TOOLTIP, data)
- #define **EWL_ATTACH_TOOLTIP_TYPE** "tooltip"
- #define **ewl_attach_tooltip_widget_set**(w, data) ewl_attach_widget_set(EWL_WIDGET(w), EWL_ATTACH_TYPE_TOOLTIP, data)
- #define **ewl_attach_widget_association_get**(w) ewl_attach_get(EWL_WIDGET(w), EWL_ATTACH_TYPE_WIDGET_ASSOCIATION)
- #define **ewl_attach_widget_association_set**(w, data)

Typedefs

- typedef **Ewl_Attach** Ewl_Attach
- typedef **Ewl_Attach_Dnd** Ewl_Attach_Dnd

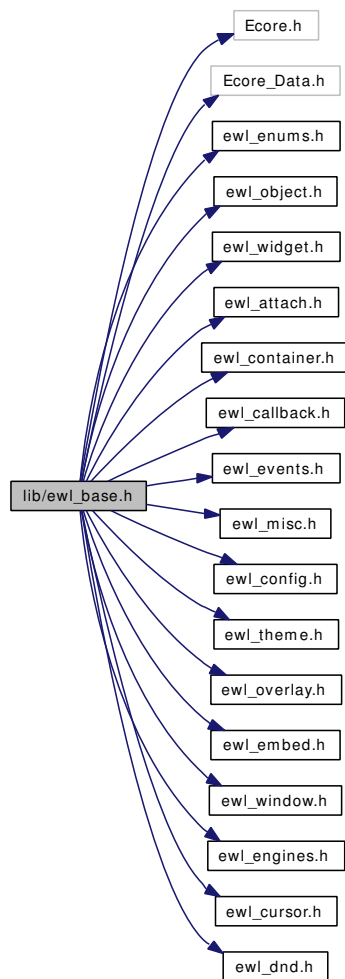
Functions

- `void * ewl_attach_get (Ewl_Widget *w, Ewl_Attach_Type t)`
Get the attachment of type `t` from the widget `w`.
- `void ewl_attach_list_del (Ewl_Attach_List *list, Ewl_Attach_Type type)`
Deletes the given type `type` from the list `list`.
- `void ewl_attach_other_set (Ewl_Widget *w, Ewl_Attach_Type t, void *data)`
Attaches the data `data` to the widget `w` with the attache type of `t`.
- `void ewl_attach_text_set (Ewl_Widget *w, Ewl_Attach_Type t, const char *data)`
Attaches the text `data` to the widget `w`.
- `void ewl_attach_widget_set (Ewl_Widget *w, Ewl_Attach_Type t, Ewl_Widget *data)`
Attaches a widget `data` to the widget `w`.

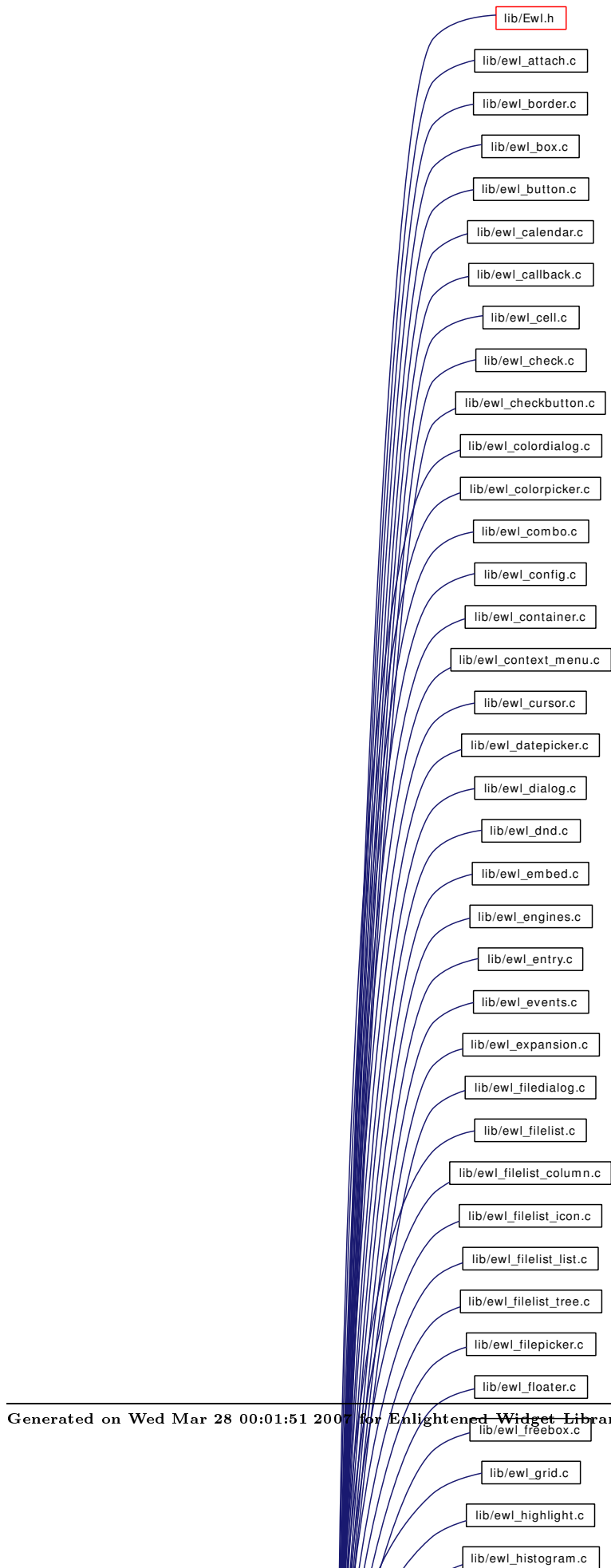
8.19 lib/ewl_base.h File Reference

```
#include <Ecore.h>
#include <Ecore_Data.h>
#include <ewl_enums.h>
#include <ewl_object.h>
#include <ewl_widget.h>
#include <ewl_attach.h>
#include <ewl_container.h>
#include <ewl_callback.h>
#include <ewl_events.h>
#include <ewl_misc.h>
#include <ewl_config.h>
#include <ewl_theme.h>
#include <ewl_overlay.h>
#include <ewl_embed.h>
#include <ewl_window.h>
#include <ewl_engines.h>
#include <ewl_cursor.h>
#include <ewl_dnd.h>
```

Include dependency graph for ewl_base.h:



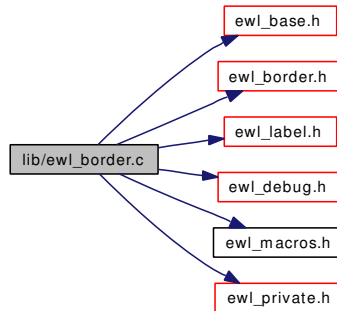
This graph shows which files directly or indirectly include this file:



8.20 lib/ewl_border.c File Reference

```
#include "ewl_base.h"
#include "ewl_border.h"
#include "ewl_label.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_border.c:



Functions

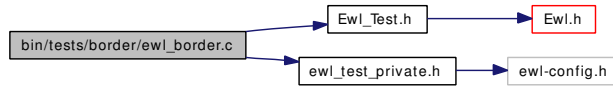
- **int ewl_border_init (Ewl_Border *b)**
Initialize a border container to default values.
- **unsigned int ewl_border_label_alignment_get (Ewl_Border *b)**
Retruns the alignment setting of the label for this border container.
- **void ewl_border_label_alignment_set (Ewl_Border *b, unsigned int align)**
alters the alignment setting of the label on the border
- **const char * ewl_border_label_get (Ewl_Border *b)**
Get the label from a border widget.
- **Ewl_Position ewl_border_label_position_get (Ewl_Border *b)**
Gets the label position of the border.
- **void ewl_border_label_position_set (Ewl_Border *b, Ewl_Position pos)**
Sets the position of the label in the border container.
- **void ewl_border_label_set (Ewl_Border *b, const char *t)**
Set the label for the border.
- **Ewl_Widget * ewl_border_new (void)**
Allocate and initialize a new border container.

8.21 bin/tests/border/ewl_border.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

Include dependency graph for ewl_border.c:



Functions

- `void test_info (Ewl_Test *test)`

8.21.1 Function Documentation

8.21.1.1 `void test_info (Ewl_Test * test)`

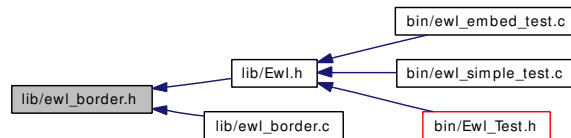
8.22 lib/ewl_border.h File Reference

```
#include "ewl_box.h"
```

Include dependency graph for ewl_border.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Border**

*Inherits from **Ewl_Box** (p. 586) to allow drawing a border and label decoration around widgets.*

Defines

- `#define EWL_BORDER(border) ((Ewl_Border *) border)`
- `#define EWL_BORDER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_BORDER_TYPE))`
- `#define EWL_BORDER_TYPE "border"`

Typedefs

- `typedef Ewl_Border Ewl_Border`

Functions

- `int ewl_border_init (Ewl_Border *b)`
Initialize a border container to default values.
- `unsigned int ewl_border_label_alignment_get (Ewl_Border *b)`
Retruns the alignment setting of the label for this border container.
- `void ewl_border_label_alignment_set (Ewl_Border *b, unsigned int align)`
alters the alignment setting of the label on the border
- `const char * ewl_border_label_get (Ewl_Border *b)`
Get the label from a border widget.
- `Ewl_Position ewl_border_label_position_get (Ewl_Border *b)`

Gets the label position of the border.

- void **ewl_border_label_position_set** (**Ewl_Border** *b, **Ewl_Position** pos)

Sets the position of the label in the border container.

- void **ewl_border_label_set** (**Ewl_Border** *b, const char *t)

Set the label for the border.

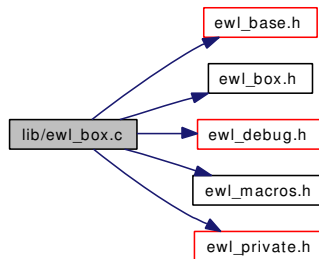
- **Ewl_Widget** * **ewl_border_new** (void)

Allocate and initialize a new border container.

8.23 lib/ewl_box.c File Reference

```
#include "ewl_base.h"
#include "ewl_box.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_box.c:



Data Structures

- struct **Ewl_Box_Orientation**

Provides information on the orientation of a box.

Functions

- void **ewl_box_cb_child_hide** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_box_cb_child_homogeneous_show** (Ewl_Container *c, Ewl_Widget *w __UNUSED__)
- void **ewl_box_cb_child_resize** (Ewl_Container *c, Ewl_Widget *w __UNUSED__, int size, Ewl_Orientation o)
- void **ewl_box_cb_child_show** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_box_cb_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_box_cb_configure_homogeneous** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_box_homogeneous_set** (Ewl_Box *b, unsigned int h)

Change the homogeneous layout of the box.

- int **ewl_box_init** (Ewl_Box *b)

Initialize the box to starting values.

- **Ewl_Widget ***ewl_box_new (void)

Allocate and initialize a new box with given orientation.

- **Ewl_Orientation** ewl_box_orientation_get (Ewl_Box *b)

Retrieves the orientation of the box.

- void `ewl_box_orientation_set` (`Ewl_Box *b`, `Ewl_Orientation o`)
Change the specified box's orientation.
- void `ewl_box_spacing_set` (`Ewl_Box *b`, `int s`)
Changes the spacing between the objects in the box.
- `Ewl_Widget *` `ewl_hbox_new` (`void`)
Allocate and initialize a new box with horizontal orientation.
- `Ewl_Widget *` `ewl_vbox_new` (`void`)
Allocate and initialize a new box with vertical orientation.

8.23.1 Function Documentation

- 8.23.1.1 void `ewl_box_cb_child_homogeneous_show` (`Ewl_Container * c`,
`Ewl_Widget *w` *UNUSED*)
- 8.23.1.2 void `ewl_box_cb_child_resize` (`Ewl_Container * c`, `Ewl_Widget *w`
UNUSED, `int size`, `Ewl_Orientation o`)
- 8.23.1.3 void `ewl_box_cb_configure` (`Ewl_Widget * w`, `void *ev_data`
UNUSED, `void *user_data` *UNUSED*)
- 8.23.1.4 void `ewl_box_cb_configure_homogeneous` (`Ewl_Widget * w`, `void`
`*ev_data` *UNUSED*, `void *user_data` *UNUSED*)

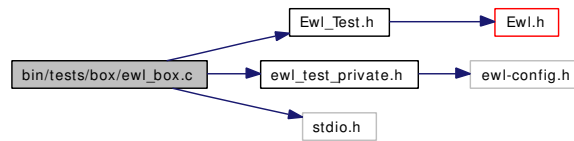
8.24 bin/tests/box/ewl_box.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_box.c:



Functions

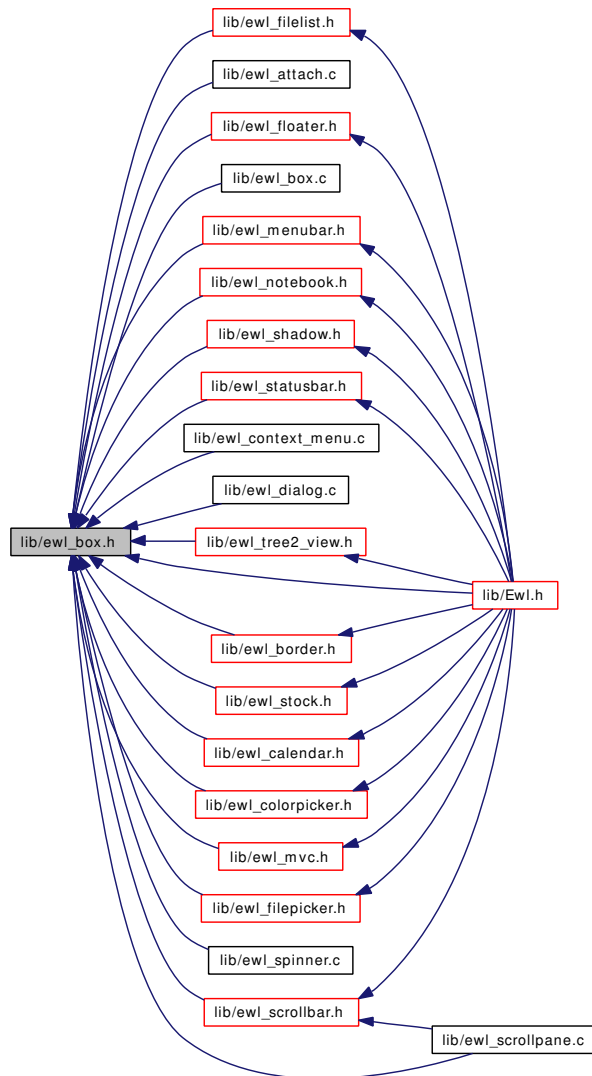
- `void test_info (Ewl_Test *test)`

8.24.1 Function Documentation

8.24.1.1 `void test_info (Ewl_Test * test)`

8.25 lib/ewl_box.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Box**

*Inherits from an **Ewl_Container** (p. 618) to provide layout facilities for child widgets placed inside.*

Defines

- `#define EWL_BOX(box) ((Ewl_Box *) box)`

- `#define EWL_BOX_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_BOX_TYPE))`
- `#define EWL_BOX_TYPE "box"`

Typedefs

- `typedef Ewl_Box Ewl_Box`

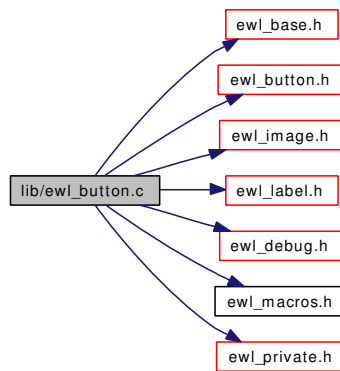
Functions

- `void ewl_box_cb_child_hide (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_box_cb_child_homogeneous_show (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_box_cb_child_resize (Ewl_Container *c, Ewl_Widget *w, int size, Ewl_Orientation o)`
- `void ewl_box_cb_child_show (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_box_cb_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_box_cb_configure_homogeneous (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_box_homogeneous_set (Ewl_Box *b, unsigned int h)`
Change the homogeneous layout of the box.
- `int ewl_box_init (Ewl_Box *box)`
Initialize the box to starting values.
- `Ewl_Widget * ewl_box_new (void)`
Allocate and initialize a new box with given orientation.
- `Ewl_Orientation ewl_box_orientation_get (Ewl_Box *b)`
Retrieves the orientation of the box.
- `void ewl_box_orientation_set (Ewl_Box *b, Ewl_Orientation o)`
Change the specified box's orientation.
- `void ewl_box_spacing_set (Ewl_Box *b, int spacing)`
Changes the spacing between the objects in the box.
- `Ewl_Widget * ewl_hbox_new (void)`
Allocate and initialize a new box with horizontal orientation.
- `Ewl_Widget * ewl_vbox_new (void)`
Allocate and initialize a new box with vertical orientation.

8.26 lib/ewl_button.c File Reference

```
#include "ewl_base.h"
#include "ewl_button.h"
#include "ewl_image.h"
#include "ewl_label.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_button.c:



Functions

- unsigned int **ewl_button_alignment_get** (Ewl_Button *b)
Gets the alignment of the contents of the button.
- void **ewl_button_alignment_set** (Ewl_Button *b, unsigned int align)
Set the alignment of the contents of the button.
- void **ewl_button_cb_key_down** (Ewl_Widget *w, void *ev, void *data __-UNUSED__)
- unsigned int **ewl_button_fill_policy_get** (Ewl_Button *b)
Retrieves the current fill policy set on the button.
- void **ewl_button_fill_policy_set** (Ewl_Button *b, unsigned int fill)
Set the fill policy of the contents of the button.
- const char * **ewl_button_image_get** (Ewl_Button *b)
Get the image file used by the button.
- void **ewl_button_image_set** (Ewl_Button *b, const char *file, const char *key)
Set the image file file and the image key key to use.
- void **ewl_button_image_size_get** (Ewl_Button *b, int *width, int *height)

Get the size of the image inside the button.

- void **ewl_button_image_size_set** (**Ewl_Button** *b, int width, int height)

Set the size of the image inside the button.

- int **ewl_button_init** (**Ewl_Button** *b)

Initialize a button to starting values.

- const char * **ewl_button_label_get** (**Ewl_Button** *b)

Retrieve the label of the specified button.

- void **ewl_button_label_set** (**Ewl_Button** *b, const char *l)

Change the label of the specified button.

- **Ewl_Widget** * **ewl_button_new** (void)

Allocate and initialize a new button.

- **Ewl_View** * **ewl_button_view_get** (void)

*Creates and returns a view to be used by **Ewl_Button** (p. 589) widgets.*

8.26.1 Function Documentation

8.26.1.1 void ewl_button_cb_key_down (**Ewl_Widget** * w, void * ev, void *data __ *UNUSED* __)

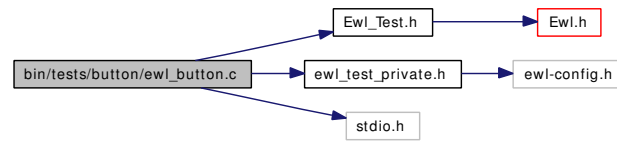
8.27 bin/tests/button/ewl_button.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_button.c:



Functions

- `void test_info (Ewl_Test *test)`

8.27.1 Function Documentation

8.27.1.1 `void test_info (Ewl_Test * test)`

8.28 lib/ewl_button.h File Reference

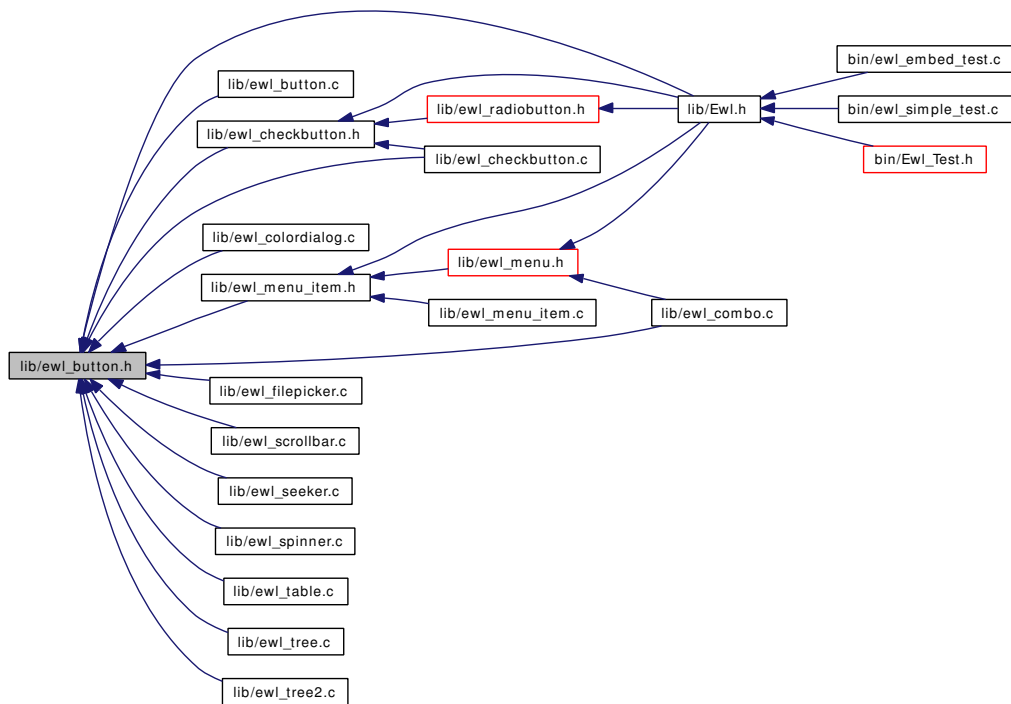
```
#include "ewl_stock.h"
```

```
#include "ewl_view.h"
```

Include dependency graph for ewl_button.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Button**

*A simple **Ewl_Widget** (p. 785) to provide for a clickable button in the UI.*

Defines

- `#define EWL_BUTTON(button) ((Ewl_Button *) button)`
- `#define EWL_BUTTON_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_BUTTON_TYPE))`
- `#define EWL_BUTTON_TYPE "button"`

Typedefs

- typedef **Ewl_Button** **Ewl_Button**

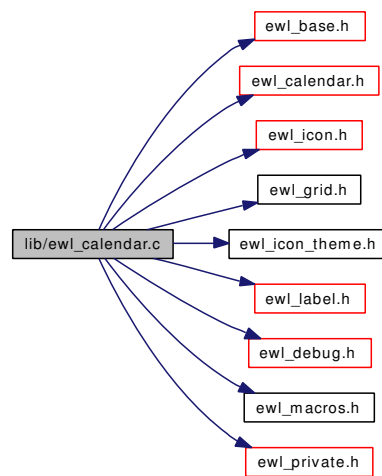
Functions

- unsigned int **ewl_button_alignment_get** (**Ewl_Button** *b)
Gets the alignment of the contents of the button.
- void **ewl_button_alignment_set** (**Ewl_Button** *b, unsigned int align)
Set the alignment of the contents of the button.
- void **ewl_button_cb_key_down** (**Ewl_Widget** *w, void *ev, void *data)
- unsigned int **ewl_button_fill_policy_get** (**Ewl_Button** *b)
Retrieves the current file policy set on the button.
- void **ewl_button_fill_policy_set** (**Ewl_Button** *b, unsigned int fill)
Set the fill policy of the contents of the button.
- const char * **ewl_button_image_get** (**Ewl_Button** *b)
Get the image file used by the button.
- void **ewl_button_image_set** (**Ewl_Button** *b, const char *file, const char *key)
*Set the image file **file** and the image key **key** to use.*
- void **ewl_button_image_size_get** (**Ewl_Button** *b, int *width, int *height)
Get the size of the image inside the button.
- void **ewl_button_image_size_set** (**Ewl_Button** *b, int width, int height)
Set the size of the image inside the button.
- int **ewl_button_init** (**Ewl_Button** *b)
Initialize a button to starting values.
- const char * **ewl_button_label_get** (**Ewl_Button** *b)
Retrieve the label of the specified button.
- void **ewl_button_label_set** (**Ewl_Button** *b, const char *l)
Change the label of the specified button.
- **Ewl_Widget** * **ewl_button_new** (void)
Allocate and initialize a new button.
- **Ewl_View** * **ewl_button_view_get** (void)
*Creates and returns a view to be used by **Ewl_Button** (p. 589) widgets.*

8.29 lib/ewl_calendar.c File Reference

```
#include "ewl_base.h"
#include "ewl_calendar.h"
#include "ewl_icon.h"
#include "ewl_grid.h"
#include "ewl_icon_theme.h"
#include "ewl_label.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_calendar.c:



Functions

- `char * ewl_calendar_ascii_time_get (Ewl_Calendar *cal)`
Returns an ASCII formatted representation of the selected date the user must free this string.
- `int ewl_calendar_day_get (Ewl_Calendar *c)`
Returns the current day selected in the calendar.
- `int ewl_calendar_init (Ewl_Calendar *ib)`
Init a new *Ewl_Calendar* (p. 591) to default values and callbacks, and set date to today.
- `int ewl_calendar_month_get (Ewl_Calendar *c)`
Returns the month selected in the calendar.
- `Ewl_Widget * ewl_calendar_new (void)`
Creates a new *Ewl_Calendar* (p. 591).

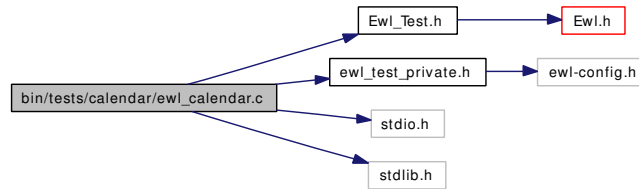
- int ewl_calendar_year_get (Ewl_Calendar *c)

Returns the year currently selected in the calendar.

8.30 bin/tests/calendar/ewl_calendar.c File Reference

```
#include "Ewl_Test.h"  
#include "ewl_test_private.h"  
#include <stdio.h>  
#include <stdlib.h>
```

Include dependency graph for ewl_calendar.c:



Functions

- void `test_info` (Ewl_Test *test)

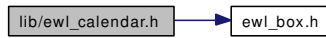
8.30.1 Function Documentation

8.30.1.1 void `test_info` (Ewl_Test * *test*)

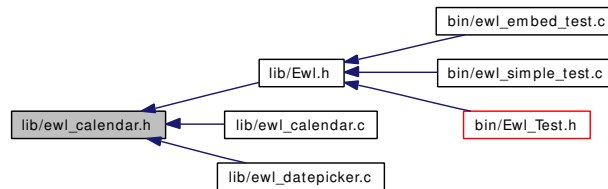
8.31 lib/ewl_calendar.h File Reference

```
#include "ewl_box.h"
```

Include dependency graph for ewl_calendar.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Calendar**

*Inherits from an **Ewl_Box** (p. 586) to provide a calendar widget.*

Defines

- #define **EWL_CALEDAR**(calendar) ((**Ewl_Calendar** *) calendar)
- #define **EWL_CALEDAR_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_CALEDAR_TYPE**))
- #define **EWL_CALEDAR_TYPE** "calendar"

Typedefs

- typedef **Ewl_Calendar** **Ewl_Calendar**

Functions

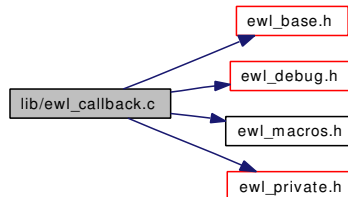
- char * **ewl_calendar_ascii_time_get** (**Ewl_Calendar** *cal)
Returns an ASCII formatted representation of the selected date the user must free this string.
- int **ewl_calendar_day_get** (**Ewl_Calendar** *c)
Returns the current day selected in the calendar.
- int **ewl_calendar_init** (**Ewl_Calendar** *calendar)
*Init a new **Ewl_Calendar** (p. 591) to default values and callbacks, and set date to today.*
- int **ewl_calendar_month_get** (**Ewl_Calendar** *c)
Returns the month selected in the calendar.

- `Ewl_Widget * ewl_calendar_new (void)`
*Creates a new **Ewl_Calendar** (p. 591).*
- `int ewl_calendar_year_get (Ewl_Calendar *c)`
Returns the year currently selected in the calendar.

8.32 lib/ewl_callback.c File Reference

```
#include "ewl_base.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_callback.c:



Functions

- **int ewl_callback_append** (Ewl_Widget *w, unsigned int t, Ewl_Callback_Function f, void *user_data)
Append a callback of the specified type.
- **void ewl_callback_call** (Ewl_Widget *w, unsigned int t)
Execute callbacks of specified types for the widget.
- **void ewl_callback_call_with_event_data** (Ewl_Widget *w, unsigned int t, void *ev_data)
Execute callbacks with event data.
- **void ewl_callback_clear** (Ewl_Widget *w)
- **void ewl_callback_del** (Ewl_Widget *w, unsigned int t, Ewl_Callback_Function f)
Delete the specified callback function from the widget.
- **void ewl_callback_del_cb_id** (Ewl_Widget *w, unsigned int t, int cb_id)
- **void ewl_callback_del_type** (Ewl_Widget *w, unsigned int t)
- **void ewl_callback_del_with_data** (Ewl_Widget *w, unsigned int t, Ewl_Callback_Function f, void *d)
Delete the specified callback function from the widget.
- **int ewl_callback_insert_after** (Ewl_Widget *w, unsigned int t, Ewl_Callback_Function f, void *user_data, Ewl_Callback_Function after, void *after_data)
Add a callback after a previous callback in list.
- **int ewl_callback_prepend** (Ewl_Widget *w, unsigned int t, Ewl_Callback_Function f, void *user_data)
prepend a callback of the specified type

- unsigned int **ewl_callback_type_add** (void)
Creates and returns a new callback identifier.
- int **ewl_callbacks_init** (void)
Setup internal registration variables for callbacks.
- void **ewl_callbacks_shutdown** (void)
Destroy internal registration variables for callbacks.

8.32.1 Function Documentation

8.32.1.1 void ewl_callback_del_cb_id (Ewl_Widget * *w*, unsigned int *t*, int *cb_id*)

8.33 lib/ewl_callback.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Callback**
Contains information about a callback.
- struct **Ewl_Callback_Custom**
Contains information about custom callbacks.

Defines

- #define **EWL_CALLBACK**(callback) ((**Ewl_Callback** *) callback)
- #define **EWL_CALLBACK_CUSTOM**(cb) ((**Ewl_Callback_Custom** *)cb)
- #define **EWL_CALLBACK_FLAG_INTERCEPT**(w, t) w → callbacks[EWL_CALLBACK_INDEX(t)].mask |= EWL_CALLBACK_NOTIFY_INTERCEPT
- #define **EWL_CALLBACK_FLAG_NOINTERCEPT**(w, t)
- #define **EWL_CALLBACK_FLAG_NOTIFY**(w, t) w → callbacks[EWL_CALLBACK_INDEX(t)].mask |= EWL_CALLBACK_NOTIFY_NOTIFY
- #define **EWL_CALLBACK_FLAGS**(w, t) (w → callbacks[EWL_CALLBACK_INDEX(t)].mask)
- #define **EWL_CALLBACK_FUNCTION**(cb_func) ((**Ewl_Callback_Function**) cb_func)
- #define **EWL_CALLBACK_INDEX**(t) ((t < EWL_CALLBACK_MAX) ? t : EWL_CALLBACK_MAX)
- #define **EWL_CALLBACK_LEN**(w, t) (w → callbacks[EWL_CALLBACK_INDEX(t)].len)
- #define **EWL_CALLBACK_LIST**(w, t) (w → callbacks[EWL_CALLBACK_INDEX(t)].list)
- #define **EWL_CALLBACK_NOTIFY_MASK** (0x3)
- #define **EWL_CALLBACK_POS**(w, t) w → callbacks[EWL_CALLBACK_INDEX(t)].index
- #define **EWL_CALLBACK_SET_DIRECT**(w, t) w → callbacks[EWL_CALLBACK_INDEX(t)].mask |= EWL_CALLBACK_TYPE_DIRECT
- #define **EWL_CALLBACK_SET_NODIRECT**(w, t) w → callbacks[EWL_CALLBACK_INDEX(t)].mask &= ~EWL_CALLBACK_TYPE_DIRECT

Typedefs

- typedef **Ewl_Callback** **Ewl_Callback**
- typedef **Ewl_Callback_Custom** **Ewl_Callback_Custom**
- typedef void(*) **Ewl_Callback_Function** (**Ewl_Widget** *widget, void *ev_data, void *user_data)

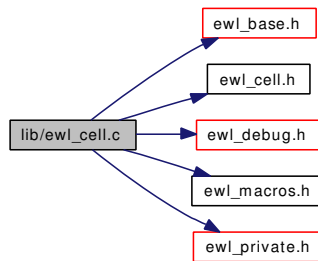
Functions

- **int ewl_callback_append** (**Ewl_Widget** *widget, unsigned int type, **Ewl_Callback_Function** func, void *user_data)
Append a callback of the specified type.
- **void ewl_callback_call** (**Ewl_Widget** *widget, unsigned int type)
Execute callbacks of specified types for the widget.
- **void ewl_callback_call_with_event_data** (**Ewl_Widget** *widget, unsigned int type, void *event_data)
Execute callbacks with event data.
- **void ewl_callback_clear** (**Ewl_Widget** *widget)
- **void ewl_callback_del** (**Ewl_Widget** *w, unsigned int t, **Ewl_Callback_Function** f)
Delete the specified callback function from the widget.
- **void ewl_callback_del_type** (**Ewl_Widget** *w, unsigned int t)
- **void ewl_callback_del_with_data** (**Ewl_Widget** *w, unsigned int t, **Ewl_Callback_Function** f, void *data)
Delete the specified callback function from the widget.
- **int ewl_callback_insert_after** (**Ewl_Widget** *w, unsigned int t, **Ewl_Callback_Function** f, void *user_data, **Ewl_Callback_Function** after, void *after_data)
Add a callback after a previous callback in list.
- **int ewl_callback_prepend** (**Ewl_Widget** *widget, unsigned int type, **Ewl_Callback_Function** func, void *user_data)
prepend a callback of the specified type
- **unsigned int ewl_callback_type_add** (void)
Creates and returns a new callback identifier.
- **int ewl_callbacks_init** (void)
Setup internal registration variables for callbacks.
- **void ewl_callbacks_shutdown** (void)
Destroy internal registration variables for callbacks.

8.34 lib/ewl_cell.c File Reference

```
#include "ewl_base.h"
#include "ewl_cell.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_cell.c:



Functions

- void ewl_cell_cb_child_resize (Ewl_Container *c, Ewl_Widget *w, int size __UNUSED__, Ewl_Orientation o __UNUSED__)
- void ewl_cell_cb_child_show (Ewl_Container *c, Ewl_Widget *w)
- void ewl_cell_cb_configure (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- int ewl_cell_init (Ewl_Cell *cell)

Initialize the cell fields of an inheriting object.

- Ewl_Widget * ewl_cell_new (void)

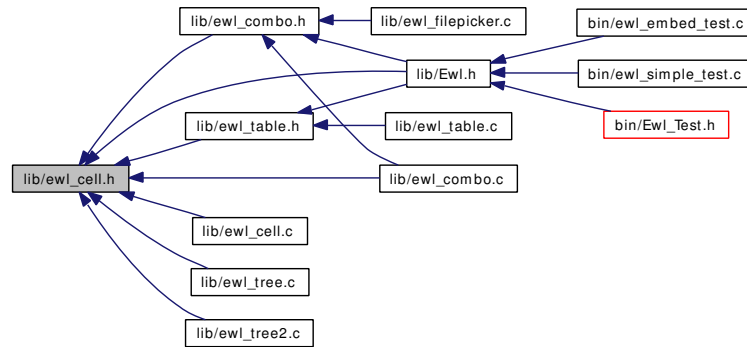
Allocate and initialize a new cell.

8.34.1 Function Documentation

8.34.1.1 void ewl_cell_cb_configure (Ewl_Widget * w, void *ev_data __UNUSED__, void *user_data __UNUSED__)

8.35 lib/ewl_cell.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Cell**

*Inherits from **Ewl_Container** (p. 618) for packing widgets inside a cell.*

Defines

- `#define EWL_CELL(t) ((Ewl_Cell *)t)`
- `#define EWL_CELL_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_CELL_TYPE))`
- `#define EWL_CELL_TYPE "cell"`

Typedefs

- `typedef Ewl_Cell Ewl_Cell`

Functions

- `void ewl_cell_cb_child_resize (Ewl_Container *c, Ewl_Widget *w, int size, Ewl_Orientation o)`
- `void ewl_cell_cb_child_show (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_cell_cb_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `int ewl_cell_init (Ewl_Cell *cell)`

Initialize the cell fields of an inheriting object.

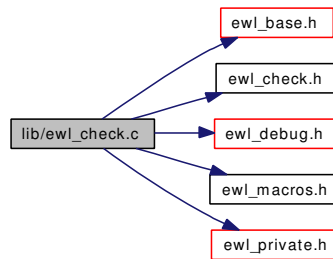
- `Ewl_Widget * ewl_cell_new (void)`

Allocate and initialize a new cell.

8.36 lib/ewl_check.c File Reference

```
#include "ewl_base.h"
#include "ewl_check.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_check.c:



Functions

- void **ewl_check_cb_clicked** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_check_cb_update_check** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_check_checked_set** (Ewl_Check *cb, int c)
Change the checked status of the check.
- int **ewl_check_init** (Ewl_Check *cb)
Initialize the members and callbacks of a check.
- int **ewl_check_is_checked** (Ewl_Check *cb)
Determine the check state of the check.
- Ewl_Widget * **ewl_check_new** (void)
Allocate and initialize a new check.

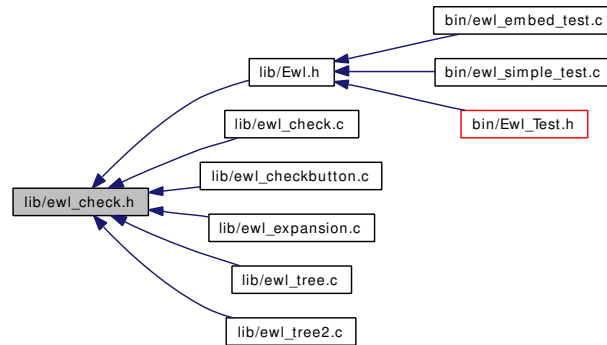
8.36.1 Function Documentation

8.36.1.1 void **ewl_check_cb_clicked** (Ewl_Widget * w, void *ev_data __UNUSED__, void *user_data __UNUSED__)

8.36.1.2 void **ewl_check_cb_update_check** (Ewl_Widget * w, void *ev_data __UNUSED__, void *user_data __UNUSED__)

8.37 lib/ewl_check.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Check**

*Inherits from **Ewl_Widget** (p. 785) and extends to provide a checked state.*

Defines

- `#define EWL_CHECK(c) ((Ewl_Check *)c)`
- `#define EWL_CHECK_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_CHECK_TYPE))`
- `#define EWL_CHECK_TYPE "check"`

Typedefs

- `typedef Ewl_Check Ewl_Check`

Functions

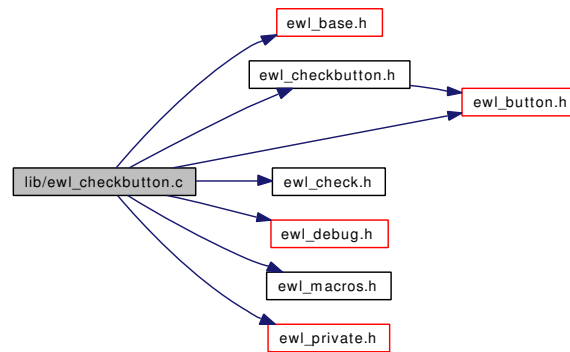
- `void ewl_check_cb_clicked (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_check_cb_update_check (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_check_checked_set (Ewl_Check *c, int checked)`
Change the checked status of the check.
- `int ewl_check_init (Ewl_Check *c)`
Initialize the members and callbacks of a check.
- `int ewl_check_is_checked (Ewl_Check *c)`
Determine the check state of the check.
- `Ewl_Widget * ewl_check_new (void)`

Allocate and initialize a new check.

8.38 lib/ewl_checkbutton.c File Reference

```
#include "ewl_base.h"
#include "ewl_checkbutton.h"
#include "ewl_button.h"
#include "ewl_check.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_checkbutton.c:



Functions

- void **ewl_checkbutton_cb_clicked** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_checkbutton_checked_set** (Ewl_Checkbutton *cb, int c)
Update the checked state of the check button.
- int **ewl_checkbutton_init** (Ewl_Checkbutton *cb)
Initialize the members and callbacks of a check button.
- int **ewl_checkbutton_is_checked** (Ewl_Checkbutton *cb)
Retrieve the checked state of a check button.
- void **ewl_checkbutton_label_position_set** (Ewl_Checkbutton *cb, Ewl_Position p)
Set the check buttons label position.
- Ewl_Widget * **ewl_checkbutton_new** (void)
Allocate and initialize a new check button.

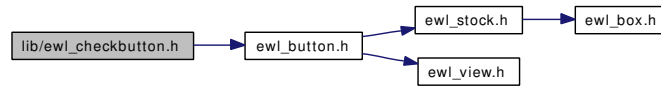
8.38.1 Function Documentation

8.38.1.1 void ewl_checkbutton_cb_clicked (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)

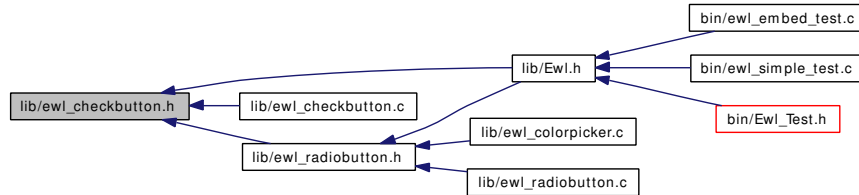
8.39 lib/ewl_checkbutton.h File Reference

```
#include "ewl_button.h"
```

Include dependency graph for ewl_checkbutton.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Checkbutton**

*Inherits from **Ewl_Button** (p. 589) and expands to provide a stateful check button.*

Defines

- `#define EWL_CHECKBUTTON(button) ((Ewl_Checkbutton *) button)`
- `#define EWL_CHECKBUTTON_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_CHECKBUTTON_TYPE))`
- `#define EWL_CHECKBUTTON_TYPE "checkboxbutton"`

Typedefs

- `typedef Ewl_Checkbutton Ewl_Checkbutton`

Functions

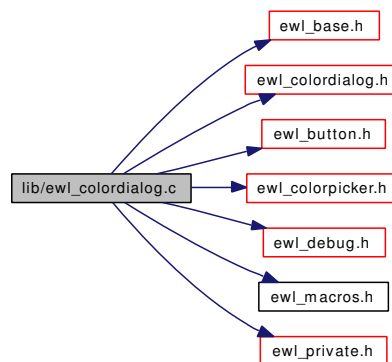
- `void ewl_checkbutton_cb_clicked (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_checkbutton_checked_set (Ewl_Checkbutton *cb, int checked)`
Update the checked state of the check button.
- `int ewl_checkbutton_init (Ewl_Checkbutton *cb)`
Initialize the members and callbacks of a check button.
- `int ewl_checkbutton_is_checked (Ewl_Checkbutton *cb)`
Retrieve the checked state of a check button.

- void ewl_checkbutton_label_position_set (Ewl_Checkbutton *cb, Ewl_Position p)
Set the check buttons label position.
- Ewl_Widget * ewl_checkbutton_new (void)
Allocate and initialize a new check button.

8.40 lib/ewl_colordialog.c File Reference

```
#include "ewl_base.h"
#include "ewl_colordialog.h"
#include "ewl_button.h"
#include "ewl_colorpicker.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_colordialog.c:



Functions

- unsigned int **ewl_colordialog_alpha_get** (Ewl_Colordialog *cd)
This retrieves the current alpha value selected in the dialog.
- void **ewl_colordialog_alpha_set** (Ewl_Colordialog *cd, unsigned int alpha)
Sets the alpha value of the cd dialog to alpha.
- void **ewl_colordialog_cb_button_click** (Ewl_Widget *w, void *ev __UNUSED__, void *data)
- void **ewl_colordialog_cb_delete_window** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- Ewl_Color_Mode **ewl_colordialog_color_mode_get** (Ewl_Colordialog *cd)
This will retrieve the current Ewl_Color_Mode set on the color dialog.
- void **ewl_colordialog_color_mode_set** (Ewl_Colordialog *cd, Ewl_Color_Mode type)
Sets the color dialog mode to type.
- void **ewl_colordialog_current_rgb_get** (Ewl_Colordialog *cd, unsigned int *r, unsigned int *g, unsigned int *b)
This will retrieve the rgb values from the color dialog.

- void **ewl_colordialog_current_rgb_set** (**Ewl_Colordialog** *cd, unsigned int r, unsigned int g, unsigned int b)
This sets the given rgb information into the color dialog.
- unsigned int **ewl_colordialog_has_alpha_get** (**Ewl_Colordialog** *cd)
This retrieves whether or not the color dialog is showing alpha information.
- void **ewl_colordialog_has_alpha_set** (**Ewl_Colordialog** *cd, unsigned int alpha)
This will set if the color dialog will show the alpha information.
- int **ewl_colordialog_init** (**Ewl_Colordialog** *cd)
Initialize a color dialog to starting values.
- **Ewl_Widget** * **ewl_colordialog_new** (void)
*Create a new **Ewl_Colordialog** (p. 601) widget.*
- void **ewl_colordialog_previous_rgb_get** (**Ewl_Colordialog** *cd, unsigned int *r, unsigned int *g, unsigned int *b)
This will retrieve the value current set into the previous color box of the color dialog rgb values from the color dialog.
- void **ewl_colordialog_previous_rgb_set** (**Ewl_Colordialog** *cd, unsigned int r, unsigned int g, unsigned int b)
This sets the given rgb values as the color to display in the previous color box of the color dialog.

8.40.1 Function Documentation

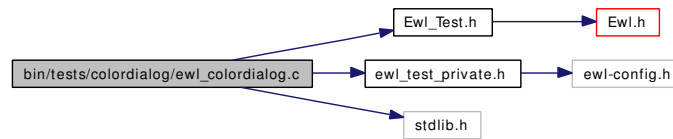
8.40.1.1 void **ewl_colordialog_cb_button_click** (**Ewl_Widget** * w, void *ev __ **UNUSED** __, void * data)

8.40.1.2 void **ewl_colordialog_cb_delete_window** (**Ewl_Widget** * w, void *ev __ **UNUSED** __, void *data __ **UNUSED** __)

8.41 bin/tests/colordialog/ewl_colordialog.c File Reference

```
#include "Ewl_Test.h"  
#include "ewl_test_private.h"  
#include <stdlib.h>
```

Include dependency graph for ewl_colordialog.c:



Functions

- `void test_info (Ewl_Test *test)`

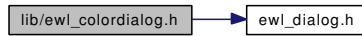
8.41.1 Function Documentation

8.41.1.1 `void test_info (Ewl_Test * test)`

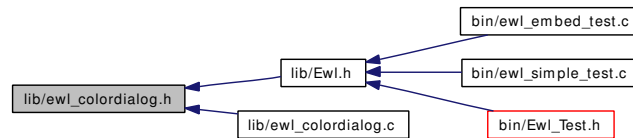
8.42 lib/ewl_colordialog.h File Reference

```
#include "ewl_dialog.h"
```

Include dependency graph for ewl_colordialog.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Colordialog**

*Inherit from the **Ewl_Dialog** (p. 626) to create a color dialog widget.*

Defines

- #define **EWL_COLORDIALOG**(cd) ((**Ewl_Colordialog** *) cd)
- #define **EWL_COLORDIALOG_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_COLORDIALOG_TYPE))
- #define **EWL_COLORDIALOG_TYPE** "colordialog"

Typedefs

- typedef **Ewl_Colordialog** **Ewl_Colordialog**

Functions

- unsigned int **ewl_colordialog_alpha_get** (**Ewl_Colordialog** *cp)
This retrieves the current alpha value selected in the dialog.
- void **ewl_colordialog_alpha_set** (**Ewl_Colordialog** *cp, unsigned int alpha)
Sets the alpha value of the cd dialog to alpha.
- void **ewl_colordialog_cb_button_click** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_colordialog_cb_delete_window** (**Ewl_Widget** *w, void *ev, void *data)
- **Ewl_Color_Mode** **ewl_colordialog_color_mode_get** (**Ewl_Colordialog** *cp)
This will retrieve the current Ewl_Color_Mode set on the color dialog.
- void **ewl_colordialog_color_mode_set** (**Ewl_Colordialog** *cp, **Ewl_Color_Mode** type)

Sets the color dialog mode to `type`.

- void **ewl_colordialog_current_rgb_get** (**Ewl_Colordialog** *cp, unsigned int *r, unsigned int *g, unsigned int *b)

This will retrieve the rgb values from the color dialog.

- void **ewl_colordialog_current_rgb_set** (**Ewl_Colordialog** *cp, unsigned int r, unsigned int g, unsigned int b)

This sets the given rgb information into the color dialog.

- unsigned int **ewl_colordialog_has_alpha_get** (**Ewl_Colordialog** *cp)

This retrieves whether or not the color dialog is showing alpha information.

- void **ewl_colordialog_has_alpha_set** (**Ewl_Colordialog** *cp, unsigned int alpha)

This will set if the color dialog will show the alpha information.

- int **ewl_colordialog_init** (**Ewl_Colordialog** *cd)

Initialize a color dialog to starting values.

- **Ewl_Widget** * **ewl_colordialog_new** (void)

*Create a new **Ewl_Colordialog** (p. 601) widget.*

- void **ewl_colordialog_previous_rgb_get** (**Ewl_Colordialog** *cp, unsigned int *r, unsigned int *g, unsigned int *b)

This will retrieve the value current set into the previous color box of the color dialog rgb values from the color dialog.

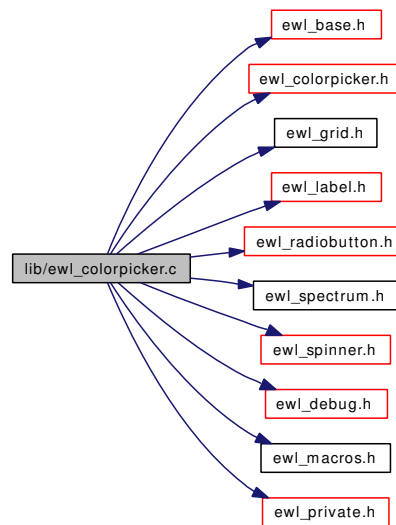
- void **ewl_colordialog_previous_rgb_set** (**Ewl_Colordialog** *cp, unsigned int r, unsigned int g, unsigned int b)

This sets the given rgb values as the color to display in the previous color box of the color dialog.

8.43 lib/ewl_colorpicker.c File Reference

```
#include "ewl_base.h"
#include "ewl_colorpicker.h"
#include "ewl_grid.h"
#include "ewl_label.h"
#include "ewl_radiobutton.h"
#include "ewl_spectrum.h"
#include "ewl_spinner.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_colorpicker.c:



Data Structures

- struct **Ewl_Colorpicker_Radiobutton**
*Inherits from **Ewl_Radiobutton** (p. 725) and extends to provide colorpicker information.*
- struct **Ewl_Colorpicker_Spinner**
*Inherits from **Ewl_Spinner** (p. 746) and extends to provide colorpicker information.*

Defines

- #define **EWL_COLORPICKER_RADIOBUTTON**(button) ((Ewl_Colorpicker_Radiobutton *)button)
- #define **EWL_COLORPICKER_RADIOBUTTON_TYPE** "colorpicker_radiobutton"

- `#define EWL_COLORPICKER_SPINNER(spinner) ((Ewl_Colorpicker_Spinner *)spinner)`
- `#define EWL_COLORPICKER_SPINNER_TYPE "colorpicker_spinner"`

Typedefs

- `typedef Ewl_Colorpicker_Radiobutton Ewl_Colorpicker_Radiobutton`
- `typedef Ewl_Colorpicker_Spinner Ewl_Colorpicker_Spinner`

Functions

- `unsigned int ewl_colorpicker_alpha_get (Ewl_Colorpicker *cp)`
Retrive the alpha value from the color picker.
- `void ewl_colorpicker_alpha_set (Ewl_Colorpicker *cp, unsigned int alpha)`
Set the alpha value into the color picker.
- `void ewl_colorpicker_cb_dnd_data (Ewl_Widget *w, void *ev, void *data __UNUSED__)`
- `void ewl_colorpicker_cb_previous_clicked (Ewl_Widget *w __UNUSED__, void *ev __UNUSED__, void *data)`
- `void ewl_colorpicker_cb_radio_change (Ewl_Widget *w, void *ev __UNUSED__, void *data)`
- `void ewl_colorpicker_cb_spinner_change (Ewl_Widget *w, void *ev __UNUSED__, void *data)`
- `void ewl_colorpicker_cb_square_change (Ewl_Widget *w, void *ev __UNUSED__, void *data)`
- `void ewl_colorpicker_cb_vertical_change (Ewl_Widget *w, void *ev __UNUSED__, void *data)`
- `Ewl_Color_Mode ewl_colorpicker_color_mode_get (Ewl_Colorpicker *cp)`
Get the color mode of the color picker.
- `void ewl_colorpicker_color_mode_set (Ewl_Colorpicker *cp, Ewl_Color_Mode mode)`
Set the color mode of the color picker.
- `void ewl_colorpicker_current_rgb_get (Ewl_Colorpicker *cp, unsigned int *r, unsigned int *g, unsigned int *b)`
Retrieve the current RGB values from the color picker.
- `void ewl_colorpicker_current_rgb_set (Ewl_Colorpicker *cp, unsigned int r, unsigned int g, unsigned int b)`
Set the current RGB values into the color picker.
- `unsigned int ewl_colorpicker_has_alpha_get (Ewl_Colorpicker *cp)`
Retrieves if the color picker displays alpha information.
- `void ewl_colorpicker_has_alpha_set (Ewl_Colorpicker *cp, unsigned int alpha)`
Set if the color picker displays alpha information.

- `int ewl_colorpicker_init (Ewl_Colorpicker *cp)`

Initializes the cp color picker widget.

- `Ewl_Widget * ewl_colorpicker_new (void)`

*Creates a new **Ewl_Colorpicker** (p. 603) widget.*

- `void ewl_colorpicker_previous_rgb_get (Ewl_Colorpicker *cp, unsigned int *r, unsigned int *g, unsigned int *b)`

Retrieve the previous RGB values from the color picker.

- `void ewl_colorpicker_previous_rgb_set (Ewl_Colorpicker *cp, unsigned int r, unsigned int g, unsigned int b)`

Set the previous RBG values into the color picker.

8.43.1 Define Documentation

8.43.1.1 `#define EWL_COLORPICKER_RADIOBUTTON(button) ((Ewl_Colorpicker_Radiobutton *)button)`

8.43.1.2 `#define EWL_COLORPICKER_RADIOBUTTON_TYPE "colorpicker_radiobutton"`

8.43.1.3 `#define EWL_COLORPICKER_SPINNER(spinner) ((Ewl_Colorpicker_Spinner *)spinner)`

8.43.1.4 `#define EWL_COLORPICKER_SPINNER_TYPE "colorpicker_spinner"`

8.43.2 Typedef Documentation

8.43.2.1 `typedef struct Ewl_Colorpicker_Radiobutton Ewl_Colorpicker_Radiobutton`

`Ewl_Colorpicker_Radiobutton` (p. 607)

8.43.2.2 `typedef struct Ewl_Colorpicker_Spinner Ewl_Colorpicker_Spinner`

`Ewl_Colorpicker_Spinner` (p. 609)

8.43.3 Function Documentation

8.43.3.1 void ewl_colorpicker_cb_dnd_data (Ewl_Widget * *w*, void * *ev*, void *data __ *UNUSED* __)

8.43.3.2 void ewl_colorpicker_cb_previous_clicked (Ewl_Widget * *w* __ *UNUSED* __, void **ev* __ *UNUSED* __, void * *data*)

8.43.3.3 void ewl_colorpicker_cb_radio_change (Ewl_Widget * *w*, void **ev* __ *UNUSED* __, void * *data*)

8.43.3.4 void ewl_colorpicker_cb_spinner_change (Ewl_Widget * *w*, void **ev* __ *UNUSED* __, void * *data*)

8.43.3.5 void ewl_colorpicker_cb_square_change (Ewl_Widget * *w*, void **ev* __ *UNUSED* __, void * *data*)

8.43.3.6 void ewl_colorpicker_cb_vertical_change (Ewl_Widget * *w*, void **ev* __ *UNUSED* __, void * *data*)

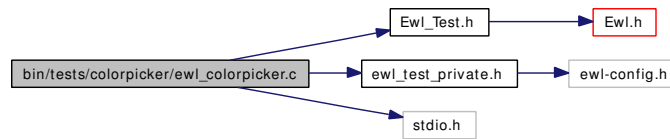
8.44 bin/tests/colorpicker/ewl_colorpicker.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_colorpicker.c:



Functions

- `void test_info (Ewl_Test *test)`

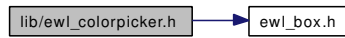
8.44.1 Function Documentation

8.44.1.1 `void test_info (Ewl_Test * test)`

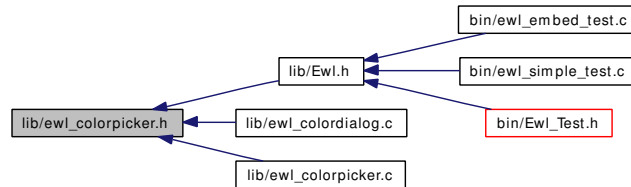
8.45 lib/ewl_colorpicker.h File Reference

```
#include "ewl_box.h"
```

Include dependency graph for ewl_colorpicker.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Colorpicker**

*Inherits from **Ewl_Box** (p. 586) to produce a color picker widget.*

Defines

- `#define EWL_COLORPICKER(cp) ((Ewl_Colorpicker *) cp)`
- `#define EWL_COLORPICKER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_COLORPICKER_TYPE))`
- `#define EWL_COLORPICKER_TYPE "colorpicker"`

Typedefs

- `typedef Ewl_Colorpicker Ewl_Colorpicker`

Functions

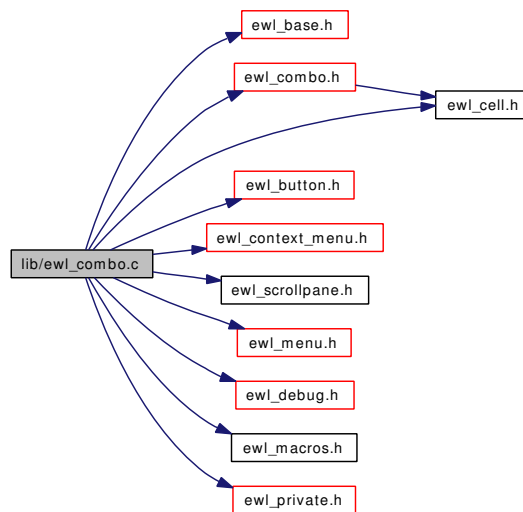
- `unsigned int ewl_colorpicker_alpha_get (Ewl_Colorpicker *cp)`
Retrive the alpha value from the color picker.
- `void ewl_colorpicker_alpha_set (Ewl_Colorpicker *cp, unsigned int alpha)`
Set the alpha value into the color picker.
- `void ewl_colorpicker_cb_dnd_data (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_colorpicker_cb_previous_clicked (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_colorpicker_cb_radio_change (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_colorpicker_cb_spinner_change (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_colorpicker_cb_square_change (Ewl_Widget *w, void *ev, void *data)`

- void **ewl_colorpicker_cb_vertical_change** (Ewl_Widget *w, void *ev, void *data)
- **Ewl_Color_Mode** **ewl_colorpicker_color_mode_get** (Ewl_Colorpicker *cp)
Get the color mode of the color picker.
- void **ewl_colorpicker_color_mode_set** (Ewl_Colorpicker *cp, Ewl_Color_Mode type)
Set the color mode of the color picker.
- void **ewl_colorpicker_current_rgb_get** (Ewl_Colorpicker *cp, unsigned int *r, unsigned int *g, unsigned int *b)
Retrieve the current RGB values from the color picker.
- void **ewl_colorpicker_current_rgb_set** (Ewl_Colorpicker *cp, unsigned int r, unsigned int g, unsigned int b)
Set the current RGB values into the color picker.
- unsigned int **ewl_colorpicker_has_alpha_get** (Ewl_Colorpicker *cp)
Retrieves if the color picker displays alpha information.
- void **ewl_colorpicker_has_alpha_set** (Ewl_Colorpicker *cp, unsigned int alpha)
Set if the color picker displays alpha information.
- int **ewl_colorpicker_init** (Ewl_Colorpicker *cp)
Initializes the cp color picker widget.
- **Ewl_Widget *** **ewl_colorpicker_new** (void)
*Creates a new **Ewl_Colorpicker** (p. 603) widget.*
- void **ewl_colorpicker_previous_rgb_get** (Ewl_Colorpicker *cp, unsigned int *r, unsigned int *g, unsigned int *b)
Retrieve the previous RGB values from the color picker.
- void **ewl_colorpicker_previous_rgb_set** (Ewl_Colorpicker *cp, unsigned int r, unsigned int g, unsigned int b)
Set the previous RGB values into the color picker.

8.46 lib/ewl_combo.c File Reference

```
#include "ewl_base.h"
#include "ewl_combo.h"
#include "ewl_button.h"
#include "ewl_cell.h"
#include "ewl_context_menu.h"
#include "ewl_scrollpane.h"
#include "ewl_menu.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_combo.c:



Functions

- void **ewl_combo_cb_decrement_clicked** (Ewl_Widget *w __UNUSED__, void *ev __UNUSED__, void *data)
- void **ewl_combo_cb_popup_hide** (Ewl_Widget *w __UNUSED__, void *ev __UNUSED__, void *data)
- void **ewl_combo_cell_cb_clicked** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- Ewl_Combo * **ewl_combo_cell_combo_get** (Ewl_Combo_Cell *cell)
- void **ewl_combo_cell_combo_set** (Ewl_Combo_Cell *cell, Ewl_Combo *combo)
- void * **ewl_combo_cell_data_get** (Ewl_Combo_Cell *cell)
- void **ewl_combo_cell_data_set** (Ewl_Combo_Cell *cell, void *mvc_data)
- int **ewl_combo_cell_init** (Ewl_Combo_Cell *cell)

Initializes a combo cell to default values.

- **Ewl_Model * ewl_combo_cell_model_get** (Ewl_Combo_Cell *cell)
- **void ewl_combo_cell_model_set** (Ewl_Combo_Cell *cell, Ewl_Model *model)
- **Ewl_Widget * ewl_combo_cell_new** (void)
Create a new combo cell.
- **unsigned int ewl_combo_editable_get** (Ewl_Combo *combo)
Retrieves the editable status of the combo.
- **void ewl_combo_editable_set** (Ewl_Combo *combo, unsigned int editable)
This will set if the displayed data in the combo is editable.
- **int ewl_combo_init** (Ewl_Combo *combo)
Initializes a combo to default values.
- **Ewl_Widget * ewl_combo_new** (void)
Create a new combo box.
- **void ewl_combo_popup_container_set** (Ewl_Combo *combo, Ewl_Container *c)
- **unsigned int ewl_combo_scrollable_get** (Ewl_Combo *combo)
Retrieves the scrollable status of the combo.
- **void ewl_combo_scrollable_set** (Ewl_Combo *combo, unsigned int scrollable)
- **Ewl_Widget * ewl_combo_submenu_new** (Ewl_Combo *c, Ewl_Model *model, Ewl_View *view, void *mvc_data)

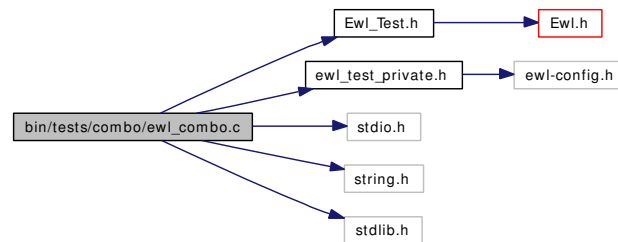
8.46.1 Function Documentation

- 8.46.1.1 void ewl_combo_cb_decrement_clicked** (Ewl_Widget *w __ *UNUSED* __, void *ev __ *UNUSED* __, void * data)
- 8.46.1.2 void ewl_combo_cb_popup_hide** (Ewl_Widget *w __ *UNUSED* __, void *ev __ *UNUSED* __, void * data)
- 8.46.1.3 void ewl_combo_cell_cb_clicked** (Ewl_Widget * w, void *ev __ *UNUSED* __, void *data __ *UNUSED* __)
- 8.46.1.4 Ewl_Widget * ewl_combo_submenu_new** (Ewl_Combo * c, Ewl_Model * model, Ewl_View * view, void * mvc_data)

8.47 bin/tests/combo/ewl_combo.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

Include dependency graph for ewl_combo.c:



Data Structures

- struct `Combo_Test_Data`

Typedefs

- typedef `Combo_Test_Data` `Combo_Test_Data`

Functions

- void `test_info` (`Ewl_Test *test`)

8.47.1 Typedef Documentation

8.47.1.1 typedef struct `Combo_Test_Data` `Combo_Test_Data`

8.47.2 Function Documentation

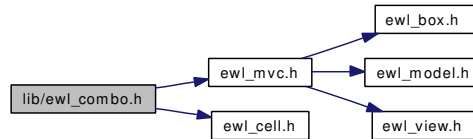
8.47.2.1 void `test_info` (`Ewl_Test *test`)

8.48 lib/ewl_combo.h File Reference

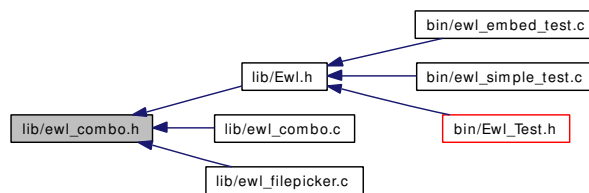
```
#include "ewl_mvc.h"
```

```
#include "ewl_cell.h"
```

Include dependency graph for ewl_combo.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Combo**

*Inherits from the **Ewl_MVC** (p. 705) widget to provide a combo widget.*

- struct **Ewl_Combo_Cell**

*Inherits from **Ewl_Cell** (p. 596) and is used internally for the combo box.*

Defines

- `#define EWL_COMBO(combo) ((Ewl_Combo *) combo)`
- `#define EWL_COMBO_CELL(c) ((Ewl_Combo_Cell *) c)`
- `#define EWL_COMBO_CELL_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_COMBO_CELL_TYPE))`
- `#define EWL_COMBO_CELL_TYPE "combo_cell"`
- `#define EWL_COMBO_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_COMBO_TYPE))`
- `#define EWL_COMBO_TYPE "combo"`

Typedefs

- `typedef Ewl_Combo Ewl_Combo`
- `typedef Ewl_Combo_Cell Ewl_Combo_Cell`

Functions

- void `ewl_combo_cb_decrement_clicked` (`Ewl_Widget *w`, void *`ev`, void *`data`)
- void `ewl_combo_cb_popup_hide` (`Ewl_Widget *w`, void *`ev`, void *`data`)
- void `ewl_combo_cell_cb_clicked` (`Ewl_Widget *w`, void *`ev`, void *`data`)
- `Ewl_Combo *``ewl_combo_cell_combo_get` (`Ewl_Combo_Cell *c`)
- void `ewl_combo_cell_combo_set` (`Ewl_Combo_Cell *c`, `Ewl_Combo *``combo`)
- void *`ewl_combo_cell_data_get` (`Ewl_Combo_Cell *c`)
- void `ewl_combo_cell_data_set` (`Ewl_Combo_Cell *c`, void *`mvc_data`)
- int `ewl_combo_cell_init` (`Ewl_Combo_Cell *c`)

Initializes a combo cell to default values.

- `Ewl_Model *``ewl_combo_cell_model_get` (`Ewl_Combo_Cell *c`)
- void `ewl_combo_cell_model_set` (`Ewl_Combo_Cell *c`, `Ewl_Model *``model`)
- `Ewl_Widget *``ewl_combo_cell_new` (void)

Create a new combo cell.

- unsigned int `ewl_combo_editable_get` (`Ewl_Combo *``combo`)

Retrieves the editable status of the combo.

- void `ewl_combo_editable_set` (`Ewl_Combo *``combo`, unsigned int `editable`)

This will set if the displayed data in the combo is editable.

- int `ewl_combo_init` (`Ewl_Combo *``combo`)

Initializes a combo to default values.

- `Ewl_Widget *``ewl_combo_new` (void)

Create a new combo box.

- void `ewl_combo_popup_container_set` (`Ewl_Combo *``combo`, `Ewl_Container *``c`)

- unsigned int `ewl_combo_scrollable_get` (`Ewl_Combo *``combo`)

Retrieves the scrollable status of the combo.

- void `ewl_combo_scrollable_set` (`Ewl_Combo *``combo`, unsigned int `scrollable`)

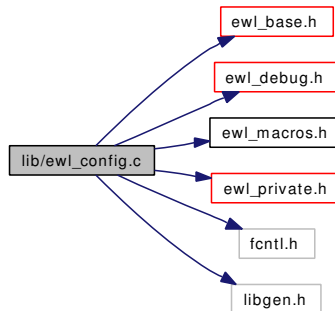
8.48.1 Function Documentation

- #### 8.48.1.1 void `ewl_combo_cell_cb_clicked` (`Ewl_Widget * w`, void * `ev`, void * `data`)

8.49 lib/ewl_config.c File Reference

```
#include "ewl_base.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
#include <fcntl.h>
#include <libgen.h>
```

Include dependency graph for ewl_config.c:



Functions

- void **ewl_config_cache_init** (void)
- int **ewl_config_can_save_system** (Ewl_Config *cfg)
Determines if the user can write the system config file.
- void **ewl_config_color_get** (Ewl_Config *cfg, const char *k, int *r, int *g, int *b, int *a)
Retrieve color associated with a key.
- void **ewl_config_color_set** (Ewl_Config *cfg, const char *k, int r, int g, int b, int a, Ewl_State_Type state)
Set the value of key to the specified color.
- void **ewl_config_destroy** (Ewl_Config *cfg)
Destroys the given config structure.
- float **ewl_config_float_get** (Ewl_Config *cfg, const char *k)
Retrieve floating point value associated with a key.
- void **ewl_config_float_set** (Ewl_Config *cfg, const char *k, float v, Ewl_State_Type state)
Set the value of key to the specified float.
- int **ewl_config_init** (void)
- int **ewl_config_int_get** (Ewl_Config *cfg, const char *k)

Retrieve integer value associated with a key.

- `void ewl_config_int_set (Ewl_Config *cfg, const char *k, int v, Ewl_State_Type state)`

Set the value of key to the specified integer.

- `Ewl_Config *ewl_config_new (const char *app_name)`

*Creates the **Ewl_Config** (p. 614) file for the given application.*

- `void ewl_config_shutdown (void)`

- `const char *ewl_config_string_get (Ewl_Config *cfg, const char *k)`

Retrieve string value associated with a key.

- `void ewl_config_string_set (Ewl_Config *cfg, const char *k, const char *v, Ewl_State_Type state)`

set the value of key to the specified string

- `int ewl_config_system_save (Ewl_Config *cfg)`

Writes out the system and user data to the system config file.

- `int ewl_config_user_save (Ewl_Config *cfg)`

Writes out the user config to the users config file.

Variables

- `Ewl_Config *ewl_config = NULL`
- `Ewl_Config_Cache ewl_config_cache`
- `Ecore_List *ewl_embed_list`

8.49.1 Variable Documentation

8.49.1.1 `Ewl_Config* ewl_config = NULL`

The system configuration data

8.49.1.2 `Ewl_Config_Cache ewl_config_cache`

system debug data

8.49.1.3 `Ecore_List* ewl_embed_list`

8.50 lib/ewl_config.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Config**
Contains Ewl configuration data.

Typedefs

- typedef **Ewl_Config** **Ewl_Config**

Functions

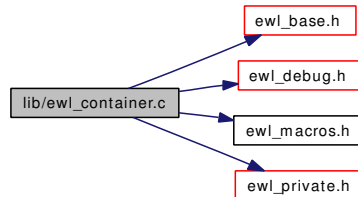
- void **ewl_config_cache_init** (void)
- int **ewl_config_can_save_system** (**Ewl_Config** *cfg)
Determines if the user can write the system config file.
- void **ewl_config_color_get** (**Ewl_Config** *cfg, const char *k, int *r, int *g, int *b, int *a)
Retrieve color associated with a key.
- void **ewl_config_color_set** (**Ewl_Config** *cfg, const char *k, int r, int g, int b, int a, **Ewl_State_Type** state)
Set the value of key to the specified color.
- void **ewl_config_destroy** (**Ewl_Config** *cfg)
Destroys the given config structure.
- float **ewl_config_float_get** (**Ewl_Config** *cfg, const char *k)
Retrieve floating point value associated with a key.
- void **ewl_config_float_set** (**Ewl_Config** *cfg, const char *k, float v, **Ewl_State_Type** state)
Set the value of key to the specified float.
- int **ewl_config_init** (void)
- int **ewl_config_int_get** (**Ewl_Config** *cfg, const char *k)
Retrieve integer value associated with a key.
- void **ewl_config_int_set** (**Ewl_Config** *cfg, const char *k, int v, **Ewl_State_Type** state)
Set the value of key to the specified integer.

- **Ewl_Config * ewl_config_new** (const char *app_name)
*Creates the **Ewl_Config** (p. 614) file for the given application.*
- void **ewl_config_shutdown** (void)
- const char * **ewl_config_string_get** (Ewl_Config *cfg, const char *k)
Retrieve string value associated with a key.
- void **ewl_config_string_set** (Ewl_Config *cfg, const char *k, const char *v, Ewl_State_Type state)
set the value of key to the specified string
- int **ewl_config_system_save** (Ewl_Config *cfg)
Writes out the system and user data to the system config file.
- int **ewl_config_user_save** (Ewl_Config *cfg)
Writes out the user config to the users config file.

8.51 lib/ewl_container.c File Reference

```
#include "ewl_base.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_container.c:



Functions

- void **ewl_container_add_notify_set** (Ewl_Container *container, Ewl_Child_Add add)
Set the function to be called when adding children.
- void **ewl_container_callback_intercept** (Ewl_Container *c, unsigned int t)
Receive a callback in place of children.
- void **ewl_container_callback_nointercept** (Ewl_Container *c, unsigned int t)
Remove a callback interception from children.
- void **ewl_container_callback_notify** (Ewl_Container *c, unsigned int t)
receive a callback of a child
- void **ewl_container_cb_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_container_cb_disable** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_container_cb_enable** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_container_cb_obscure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_container_cb_realize** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_container_cb_reparent** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_container_cb_reveal** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_container_cb_unrealize** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_container_child_add_call** (Ewl_Container *c, Ewl_Widget *w)
Triggers the child_add callback for the container c.

- **void ewl_container_child_append** (Ewl_Container *pc, Ewl_Widget *child)
Add a child at the end of the container.
- **Ewl_Widget * ewl_container_child_at_get** (Ewl_Container *widget, int x, int y)
Find the child at the given coordinates.
- **Ewl_Widget * ewl_container_child_at_recursive_get** (Ewl_Container *widget, int x, int y)
Find child that intersects coordinates.
- **int ewl_container_child_count_get** (Ewl_Container *c)
Returns the number of child widgets in the container.
- **int ewl_container_child_count_internal_get** (Ewl_Container *c)
Returns the number of child widgets in the container.
- **Ewl_Widget * ewl_container_child_get** (Ewl_Container *parent, int index)
Retrieve the child at the given index in the container.
- **void ewl_container_child_hide_call** (Ewl_Container *c, Ewl_Widget *w)
Triggers the child_hide callback for the container c.
- **int ewl_container_child_index_get** (Ewl_Container *parent, Ewl_Widget *w)
Retrieves the index of the given child in the container or -1 if not found.
- **int ewl_container_child_index_internal_get** (Ewl_Container *parent, Ewl_Widget *w)
Retrieves the index of the given child in the container taking internal widgets into account or -1 if not found.
- **void ewl_container_child_insert** (Ewl_Container *pc, Ewl_Widget *child, int index)
Add a child at an index of the container.
- **void ewl_container_child_insert_internal** (Ewl_Container *pc, Ewl_Widget *child, int index)
Add a child at an index of the container.
- **Ewl_Widget * ewl_container_child_internal_get** (Ewl_Container *parent, int index)
Retrieves the child at the given index in the container taking internal widgets into account.
- **void ewl_container_child_iterate_begin** (Ewl_Container *c)
Prepare the container to iterate over it's children.
- **void ewl_container_child_iterator_set** (Ewl_Container *c, Ewl_Container_Iterator i)
Changes the iterator function for a container.

- **Ewl_Widget * ewl_container_child_next** (Ewl_Container *c)
Retrieve the next eligible child in a container.
- **void ewl_container_child_prepend** (Ewl_Container *pc, Ewl_Widget *child)
Add a child at the beginning of the container.
- **void ewl_container_child_remove** (Ewl_Container *pc, Ewl_Widget *child)
Remove the specified child from the container.
- **void ewl_container_child_remove_call** (Ewl_Container *c, Ewl_Widget *w, int idx)
Triggers the child_remove callback for the container c.
- **void ewl_container_child_resize** (Ewl_Widget *w, int size, Ewl_Orientation o)
Notify a container of a child pref size change.
- **void ewl_container_child_show_call** (Ewl_Container *c, Ewl_Widget *w)
Triggers the child_show callback for the container c.
- **void ewl_container_destroy** (Ewl_Container *c)
Destroy all the sub-children of the container.
- **Ewl_Container * ewl_container_end_redirect_get** (Ewl_Container *c)
Searches for the last redirected container of the container.
- **void ewl_container_hide_notify_set** (Ewl_Container *container, Ewl_Child_Hide hide)
Set the function to be called when hideing children.
- **int ewl_container_init** (Ewl_Container *c)
Initialize a containers default fields and callbacks.
- **void ewl_container_largest_prefer** (Ewl_Container *c, Ewl_Orientation o)
Set preferred size to widest child.
- **Ewl_Container * ewl_container_redirect_get** (Ewl_Container *c)
Retrieves for the redirected container of the container.
- **void ewl_container_redirect_set** (Ewl_Container *c, Ewl_Container *rc)
Changes the redirected container of the container.
- **void ewl_container_remove_notify_set** (Ewl_Container *container, Ewl_Child_Remove remove)
Set the function to be called when removing children.
- **void ewl_container_reset** (Ewl_Container *c)
Destroy all children of the container.

- void `ewl_container_resize_notify_set` (`Ewl_Container *container`, `Ewl_Child_Resize` `resize`)
Set function to be called when children resize.
- void `ewl_container_show_notify_set` (`Ewl_Container *container`, `Ewl_Child_Show` `show`)
Set the function to be called when showing children.
- void `ewl_container_sum_prefer` (`Ewl_Container *c`, `Ewl_Orientation o`)
Set the sum orientation of the container.

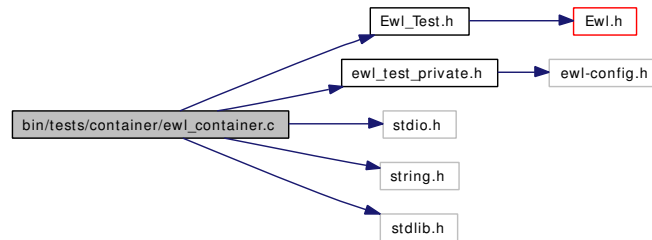
8.51.1 Function Documentation

- 8.51.1.1 void `ewl_container_cb_configure` (`Ewl_Widget * w`, void `*ev_data` __ `UNUSED` __, void `*user_data` __ `UNUSED` __)
- 8.51.1.2 void `ewl_container_cb_disable` (`Ewl_Widget * w`, void `*ev_data` __ `UNUSED` __, void `*user_data` __ `UNUSED` __)
- 8.51.1.3 void `ewl_container_cb_enable` (`Ewl_Widget * w`, void `*ev_data` __ `UNUSED` __, void `*user_data` __ `UNUSED` __)
- 8.51.1.4 void `ewl_container_cb_obscure` (`Ewl_Widget * w`, void `*ev_data` __ `UNUSED` __, void `*user_data` __ `UNUSED` __)
- 8.51.1.5 void `ewl_container_cb_realize` (`Ewl_Widget * w`, void `*ev_data` __ `UNUSED` __, void `*user_data` __ `UNUSED` __)
- 8.51.1.6 void `ewl_container_cb_reparent` (`Ewl_Widget * w`, void `*ev_data` __ `UNUSED` __, void `*user_data` __ `UNUSED` __)
- 8.51.1.7 void `ewl_container_cb_reveal` (`Ewl_Widget * w`, void `*ev_data` __ `UNUSED` __, void `*user_data` __ `UNUSED` __)
- 8.51.1.8 void `ewl_container_cb_unrealize` (`Ewl_Widget * w`, void `*ev_data` __ `UNUSED` __, void `*user_data` __ `UNUSED` __)

8.52 bin/tests/container/ewl_container.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

Include dependency graph for ewl_container.c:



Functions

- void `test_info` (Ewl_Test *test)

8.52.1 Function Documentation

8.52.1.1 void `test_info` (Ewl_Test * *test*)

8.53 lib/ewl_container.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Container**

*Inherits from the **Ewl_Widget** (p. 785) and expands to allow for placing child widgets within the available space. Also adds notifiers for various child events.*

Defines

- `#define EWL_CONTAINER(widget) ((Ewl_Container *) widget)`
*Typecast a pointer to an **Ewl_Container** (p. 618) pointer.*
- `#define EWL_CONTAINER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_CONTAINER_TYPE))`
- `#define EWL_CONTAINER_TYPE "container"`

Typedefs

- `typedef void(*) Ewl_Child_Add (Ewl_Container *c, Ewl_Widget *w)`
- `typedef void(*) Ewl_Child_Hide (Ewl_Container *c, Ewl_Widget *w)`
- `typedef void(*) Ewl_Child_Remove (Ewl_Container *c, Ewl_Widget *w, int idx)`
- `typedef void(*) Ewl_Child_Resize (Ewl_Container *c, Ewl_Widget *w, int size, Ewl_Orientation o)`
- `typedef void(*) Ewl_Child_Show (Ewl_Container *c, Ewl_Widget *w)`
- `typedef Ewl_Container Ewl_Container`
- `typedef Ewl_Widget (*)(Ewl_Container *c) Ewl_Container_Iterator (Ewl_Container *c)`

Functions

- `void ewl_container_add_notify_set (Ewl_Container *container, Ewl_Child_Add add)`
Set the function to be called when adding children.
- `void ewl_container_callback_intercept (Ewl_Container *c, unsigned int t)`
Receive a callback in place of children.
- `void ewl_container_callback_nointercept (Ewl_Container *c, unsigned int t)`
Remove a callback interception from children.
- `void ewl_container_callback_notify (Ewl_Container *c, unsigned int t)`
receive a callback of a child

- void **ewl_container_cb_configure** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_cb_disable** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_cb_enable** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_cb_obscure** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_cb_realize** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_cb_reparent** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_cb_reveal** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_cb_unrealize** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_container_child_add_call** (Ewl_Container *c, Ewl_Widget *w)
Triggers the child_add callback for the container c.
- void **ewl_container_child_append** (Ewl_Container *parent, Ewl_Widget *child)
Add a child at the end of the container.
- Ewl_Widget * **ewl_container_child_at_get** (Ewl_Container *widget, int x, int y)
Find the child at the given coordinates.
- Ewl_Widget * **ewl_container_child_at_recursive_get** (Ewl_Container *widget, int x, int y)
Find child that intersects coordinates.
- int **ewl_container_child_count_get** (Ewl_Container *c)
Returns the number of child widgets in the container.
- int **ewl_container_child_count_internal_get** (Ewl_Container *c)
Returns the number of child widgets in the container.
- Ewl_Widget * **ewl_container_child_get** (Ewl_Container *parent, int index)
Retrieve the child at the given index in the container.
- void **ewl_container_child_hide_call** (Ewl_Container *c, Ewl_Widget *w)
Triggers the child_hide callback for the container c.
- int **ewl_container_child_index_get** (Ewl_Container *parent, Ewl_Widget *child)
Retrieves the index of the given child in the container or -1 if not found.
- int **ewl_container_child_index_internal_get** (Ewl_Container *parent, Ewl_Widget *child)
Retrieves the index of the given child in the container taking internal widgets into account or -1 if not found.
- void **ewl_container_child_insert** (Ewl_Container *parent, Ewl_Widget *child, int index)
Add a child at an index of the container.

- **void ewl_container_child_insert_internal** (**Ewl_Container** *parent, **Ewl_Widget** *child, int index)
Add a child at an index of the container.
- **Ewl_Widget * ewl_container_child_internal_get** (**Ewl_Container** *parent, int index)
Retrieves the child at the given index in the container taking internal widgets into account.
- **void ewl_container_child_iterate_begin** (**Ewl_Container** *c)
Prepare the container to iterate over it's children.
- **void ewl_container_child_iterator_set** (**Ewl_Container** *c, **Ewl_Container_Iterator** i)
Changes the iterator function for a container.
- **Ewl_Widget * ewl_container_child_next** (**Ewl_Container** *c)
Retrieve the next eligible child in a container.
- **void ewl_container_child_prepend** (**Ewl_Container** *parent, **Ewl_Widget** *child)
Add a child at the beginning of the container.
- **void ewl_container_child_remove** (**Ewl_Container** *parent, **Ewl_Widget** *child)
Remove the specified child from the container.
- **void ewl_container_child_remove_call** (**Ewl_Container** *c, **Ewl_Widget** *w, int idx)
Triggers the child_remove callback for the container c.
- **void ewl_container_child_resize** (**Ewl_Widget** *w, int size, **Ewl_Orientation** o)
Notify a container of a child pref size change.
- **void ewl_container_child_show_call** (**Ewl_Container** *c, **Ewl_Widget** *w)
Triggers the child_show callback for the container c.
- **void ewl_container_destroy** (**Ewl_Container** *c)
Destroy all the sub-children of the container.
- **Ewl_Container * ewl_container_end_redirect_get** (**Ewl_Container** *c)
Searches for the last redirected container of the container.
- **void ewl_container_hide_notify_set** (**Ewl_Container** *container, **Ewl_Child_Hide** show)
Set the function to be called when hideing children.
- **int ewl_container_init** (**Ewl_Container** *container)
Initialize a containers default fields and callbacks.
- **void ewl_container_largest_prefer** (**Ewl_Container** *c, **Ewl_Orientation** o)

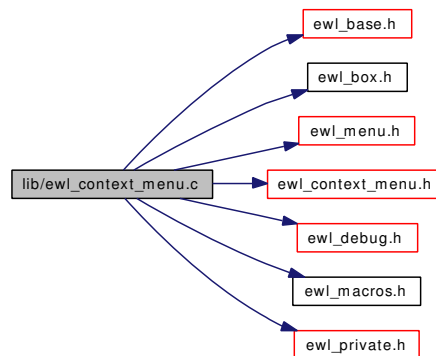
Set preferred size to widest child.

- **Ewl_Container * ewl_container_redirect_get (Ewl_Container *c)**
Retrieves for the redirected container of the container.
- **void ewl_container_redirect_set (Ewl_Container *c, Ewl_Container *rc)**
Changes the redirected container of the container.
- **void ewl_container_remove_notify_set (Ewl_Container *container, Ewl_Child_Remove remove)**
Set the function to be called when removing children.
- **void ewl_container_reset (Ewl_Container *c)**
Destroy all children of the container.
- **void ewl_container_resize_notify_set (Ewl_Container *container, Ewl_Child_Resize resize)**
Set function to be called when children resize.
- **void ewl_container_show_notify_set (Ewl_Container *container, Ewl_Child_Show show)**
Set the function to be called when showing children.
- **void ewl_container_sum_prefer (Ewl_Container *c, Ewl_Orientation o)**
Set the sum orientation of the container.

8.54 lib/ewl_context_menu.c File Reference

```
#include "ewl_base.h"
#include "ewl_box.h"
#include "ewl_menu.h"
#include "ewl_context_menu.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_context_menu.c:



Functions

- void ewl_context_menu_attach (Ewl_Context_Menu *cm, Ewl_Widget *w)
- void ewl_context_menu_cb_attach_mouse_down (Ewl_Widget *w __UNUSED__, void *ev_data, void *user_data)
- void ewl_context_menu_cb_child_add (Ewl_Container *c, Ewl_Widget *w)
- void ewl_context_menu_cb_child_clicked (Ewl_Widget *w __UNUSED__, void *ev_data __UNUSED__, void *user_data)
- void ewl_context_menu_cb_child_mouse_in (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void ewl_context_menu_cb_child_remove (Ewl_Container *c, Ewl_Widget *w, int idx)
- void ewl_context_menu_cb_focus_in (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void ewl_context_menu_cb_hide (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void ewl_context_menu_cb_mouse_down (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void ewl_context_menu_cb_mouse_move (Ewl_Widget *w, void *ev_data, void *user_data __UNUSED__)
- void ewl_context_menu_container_set (Ewl_Context_Menu *cm, Ewl_Container *c)

set a custom container for the context menu

- void ewl_context_menu_detach (Ewl_Context_Menu *cm, Ewl_Widget *w)
- int ewl_context_menu_init (Ewl_Context_Menu *cm)
Initialize the context_menu and inherited fields.
- Ewl_Widget * ewl_context_menu_new (void)
Allocate a new context_menu widget.

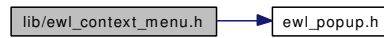
8.54.1 Function Documentation

- 8.54.1.1 void ewl_context_menu_cb_attach_mouse_down (Ewl_Widget *w
__ *UNUSED* __, void * *ev_data*, void * *user_data*)
- 8.54.1.2 void ewl_context_menu_cb_child_clicked (Ewl_Widget *w
__ *UNUSED* __, void * *ev_data* __ *UNUSED* __, void * *user_data*)
- 8.54.1.3 void ewl_context_menu_cb_child_mouse_in (Ewl_Widget * w, void
* *ev_data* __ *UNUSED* __, void * *user_data* __ *UNUSED* __)
- 8.54.1.4 void ewl_context_menu_cb_focus_in (Ewl_Widget * w, void * *ev_data*
__ *UNUSED* __, void * *user_data* __ *UNUSED* __)
- 8.54.1.5 void ewl_context_menu_cb_hide (Ewl_Widget * w, void * *ev_data*
__ *UNUSED* __, void * *user_data* __ *UNUSED* __)
- 8.54.1.6 void ewl_context_menu_cb_mouse_down (Ewl_Widget * w, void
* *ev_data* __ *UNUSED* __, void * *user_data* __ *UNUSED* __)
- 8.54.1.7 void ewl_context_menu_cb_mouse_move (Ewl_Widget * w, void *
ev_data, void * *user_data* __ *UNUSED* __)

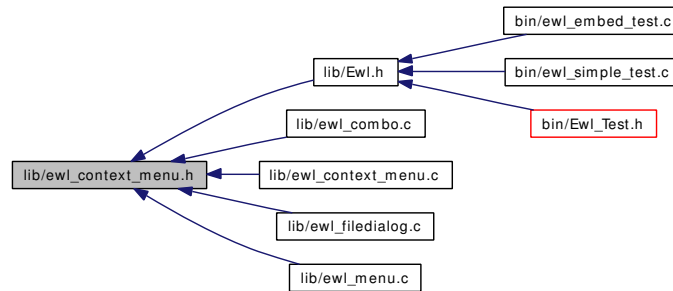
8.55 lib/ewl_context_menu.h File Reference

```
#include "ewl_popup.h"
```

Include dependency graph for ewl_context_menu.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Context_Menu**

*Inherits from **Ewl_Popup** (p. 721) and extends to provide a context menu.*

Defines

- `#define EWL_CONTEXT_MENU(context_menu) ((Ewl_Context_Menu *) context_menu)`
- `#define EWL_CONTEXT_MENU_IS(w)`
- `#define EWL_CONTEXT_MENU_TYPE "context_menu"`

Typedefs

- `typedef Ewl_Context_Menu Ewl_Context_Menu`

Functions

- `void ewl_context_menu_attach (Ewl_Context_Menu *cm, Ewl_Widget *w)`
- `void ewl_context_menu_cb_attach_mouse_down (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_context_menu_cb_child_add (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_context_menu_cb_child_clicked (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_context_menu_cb_child_mouse_in (Ewl_Widget *w, void *ev_data, void *user_data)`

- void **ewl_context_menu_cb_child_remove** (**Ewl_Container** *c, **Ewl_Widget** *w, int idx)
- void **ewl_context_menu_cb_focus_in** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_context_menu_cb_hide** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_context_menu_cb_mouse_down** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_context_menu_cb_mouse_move** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_context_menu_container_set** (**Ewl_Context_Menu** *cm, **Ewl_Container** *c)

set a custom container for the context menu

- void **ewl_context_menu_detach** (**Ewl_Context_Menu** *cm, **Ewl_Widget** *w)
- int **ewl_context_menu_init** (**Ewl_Context_Menu** *cm)

Initialize the context_menu and inherited fields.

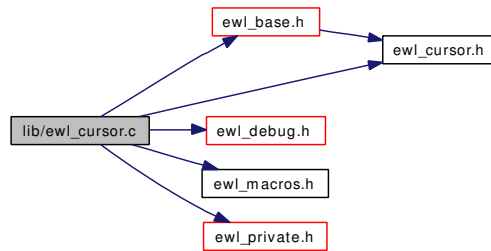
- **Ewl_Widget** * **ewl_context_menu_new** (void)

Allocate a new context_menu widget.

8.56 lib/ewl_cursor.c File Reference

```
#include "ewl_base.h"
#include "ewl_cursor.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_cursor.c:

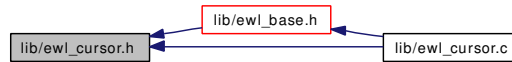


Functions

- `int ewl_cursor_init (Ewl_Cursor *cursor)`
*Initializes an **Ewl_Cursor** (p. 622) widget.*
- `Ewl_Widget * ewl_cursor_new (void)`
*Creates and initializes a new **Ewl_Cursor** (p. 622) widget.*

8.57 lib/ewl_cursor.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Cursor**

*Inherits from **Ewl_Window** (p. 788) to allow packing widgets as cursor objects.*

Defines

- #define **EWL_CURSOR**(cursor) ((**Ewl_Cursor** *) cursor)
- #define **EWL_CURSOR_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_CURSOR_TYPE))
- #define **EWL_CURSOR_TYPE** "cursor"

Typedefs

- typedef **Ewl_Cursor** **Ewl_Cursor**

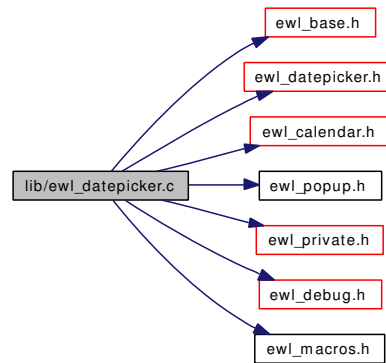
Functions

- int **ewl_cursor_init** (**Ewl_Cursor** *c)
*Initializes an **Ewl_Cursor** (p. 622) widget.*
- **Ewl_Widget** * **ewl_cursor_new** (void)
*Creates and initializes a new **Ewl_Cursor** (p. 622) widget.*

8.58 lib/ewl_datepicker.c File Reference

```
#include "ewl_base.h"
#include "ewl_datepicker.h"
#include "ewl_calendar.h"
#include "ewl_popup.h"
#include "ewl_private.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
```

Include dependency graph for ewl_datepicker.c:



Functions

- void **ewl_datepicker_cb_destroy** (Ewl_Widget *w, void *ev __UNUSED__, void *user_data __UNUSED__)
- void **ewl_datepicker_cb_value_changed** (Ewl_Widget *w __UNUSED__, void *ev __UNUSED__, void *user_data)
- void **ewl_datepicker_cb_window_mouse_down** (Ewl_Widget *w, void *ev __UNUSED__, void *user_data __UNUSED__)
- int **ewl_datepicker_init** (Ewl_Datepicker *dp)

Initialize the date picker to default values.

- Ewl_Widget * **ewl_datepicker_new** (void)

*Creates a new **Ewl_Datepicker** (p. 624).*

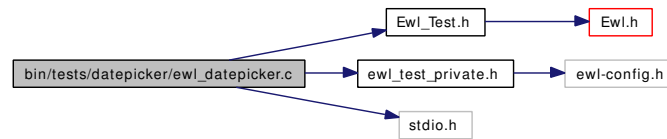
8.58.1 Function Documentation

- 8.58.1.1 void ewl_datepicker_cb_destroy (Ewl_Widget * *w*, void *ev
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.58.1.2 void ewl_datepicker_cb_value_changed (Ewl_Widget *w
__ *UNUSED* __, void *ev __ *UNUSED* __, void * *user_data*)
- 8.58.1.3 void ewl_datepicker_cb_window_mouse_down (Ewl_Widget * *w*, void
*ev __ *UNUSED* __, void *user_data __ *UNUSED* __)

8.59 bin/tests/datepicker/ewl_datepicker.c File Reference

```
#include "Ewl_Test.h"  
#include "ewl_test_private.h"  
#include <stdio.h>
```

Include dependency graph for ewl_datepicker.c:



Functions

- `void test_info (Ewl_Test *test)`

8.59.1 Function Documentation

8.59.1.1 `void test_info (Ewl_Test * test)`

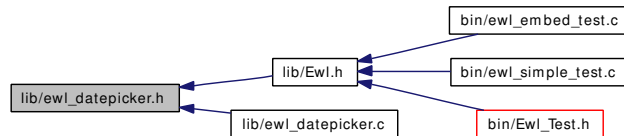
8.60 lib/ewl_datepicker.h File Reference

```
#include "ewl_text.h"
```

Include dependency graph for ewl_datepicker.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Datepicker**

*Inherits from **Ewl_Text** (p. 757) and extends to provide a date picker.*

Defines

- `#define EWL_DATEPICKER(datepicker) ((Ewl_Datepicker *) datepicker)`
- `#define EWL_DATEPICKER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_DATEPICKER_TYPE))`
- `#define EWL_DATEPICKER_TYPE "datepicker"`

Typedefs

- `typedef Ewl_Datepicker Ewl_Datepicker`

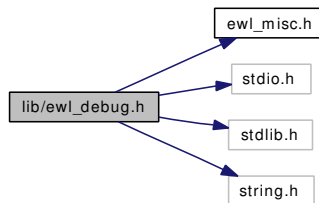
Functions

- `void ewl_datepicker_cb_destroy (Ewl_Widget *w, void *ev, void *user_data)`
- `void ewl_datepicker_cb_value_changed (Ewl_Widget *w, void *ev, void *user_data)`
- `void ewl_datepicker_cb_window_mouse_down (Ewl_Widget *w, void *ev, void *user_data)`
- `int ewl_datepicker_init (Ewl_Datepicker *datepicker)`
Initialize the date picker to default values.
- `Ewl_Widget * ewl_datepicker_new (void)`
*Creates a new **Ewl_Datepicker** (p. 624).*

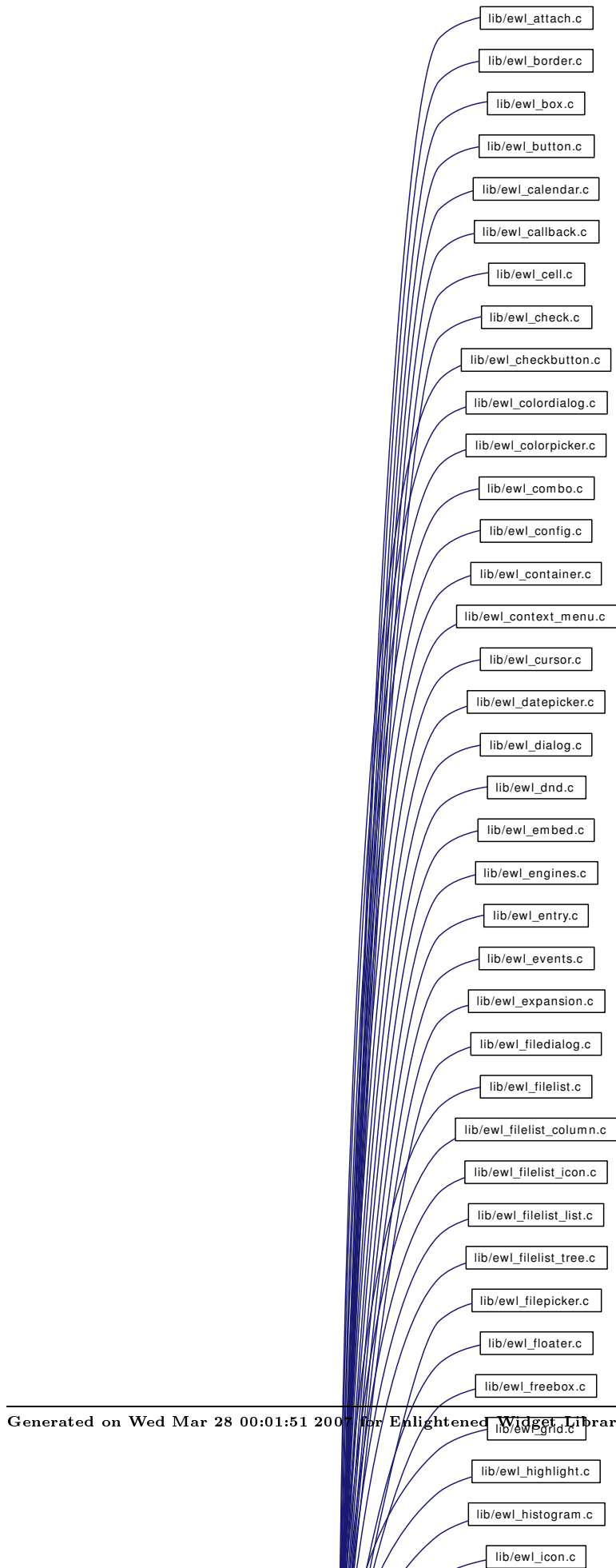
8.61 lib/ewl_debug.h File Reference

```
#include "ewl_misc.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for ewl_debug.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Config_Cache**
Contains cached Ewl configuration data.

Defines

- #define **DCHECK_PARAM_PTR**(str, ptr)
- #define **DCHECK_PARAM_PTR_RET**(str, ptr, ret)
- #define **DCHECK_TYPE**(str, ptr, type) {}
- #define **DCHECK_TYPE_RET**(str, ptr, type, ret) {}
- #define **DENTER_FUNCTION**(lvl) {}
- #define **DERROR**(fmt)
- #define **DLEAVE_FUNCTION**(lvl) {}
- #define **DLEVEL_STABLE** 20
- #define **DLEVEL_TESTING** 10
- #define **DLEVEL_UNSTABLE** 1
- #define **DRETURN**(lvl) return
- #define **DRETURN_FLOAT**(num, lvl) return num
- #define **DRETURN_INT**(num, lvl) return num
- #define **DRETURN_PTR**(ptr, lvl) return (void *) (ptr)
- #define **DWARNING**(fmt, args...) {}

Typedefs

- typedef **Ewl_Config_Cache** **Ewl_Config_Cache**

Functions

- void **ewl_backtrace** (void)
This will print a backtrace at the given point.
- void **ewl_print_warning** (void)
This is used by debugging macros for breakpoints.
- void **ewl_segv** (void)
This will cause EWL to SEGV. (Handy for debugging).

Variables

- **Ewl_Config_Cache** **ewl_config_cache**

8.61.1 Define Documentation

8.61.1.1 #define DCHECK_PARAM_PTR(str, ptr)

Value:

```
{ \
    if (!(ptr)) { \
        return; \
    } \
}
```

8.61.1.2 #define DCHECK_PARAM_PTR_RET(str, ptr, ret)

Value:

```
{ \
    if (!(ptr)) { \
        return ret; \
    } \
}
```

8.61.1.3 #define DCHECK_TYPE(str, ptr, type) {}

8.61.1.4 #define DCHECK_TYPE_RET(str, ptr, type, ret) {}

8.61.1.5 #define DENTER_FUNCTION(lvl) {}

8.61.1.6 #define DERROR(fmt)

Value:

```
{ \
    ewl_print_warning(); \
    fprintf(stderr, "\tIn function:\n\n" \
        "\t%s();\n\n", __func__); \
    fprintf(stderr, fmt); \
}
```

8.61.1.7 `#define DLEAVE_FUNCTION(lvl) {}`

8.61.1.8 `#define DLEVEL_STABLE 20`

8.61.1.9 `#define DLEVEL_TESTING 10`

8.61.1.10 `#define DLEVEL_UNSTABLE 1`

8.61.1.11 `#define DRETURN(lvl) return`

8.61.1.12 `#define DRETURN_FLOAT(num, lvl) return num`

8.61.1.13 `#define DRETURN_INT(num, lvl) return num`

8.61.1.14 `#define DRETURN_PTR(ptr, lvl) return (void*)(ptr)`

8.61.1.15 `#define DWARNING(fmt, args...) {}`

8.61.2 Typedef Documentation

8.61.2.1 `typedef struct Ewl_Config_Cache Ewl_Config_Cache`

The `Ewl_Config_Cache` (p. 615) structure

8.61.3 Function Documentation

8.61.3.1 `void ewl_backtrace (void)`

This will print a backtrace at the given point.

Returns:

Returns no value.

8.61.3.2 `void ewl_print_warning (void)`

This is used by debugging macros for breakpoints.

Returns:

Returns no value.

Set a breakpoint at this function in order to retrieve backtraces from warning messages.

8.61.3.3 `void ewl_segv (void)`

This will cause EWL to SEGV. (Handy for debugging).

Returns:

Returns no value.

8.61.4 Variable Documentation

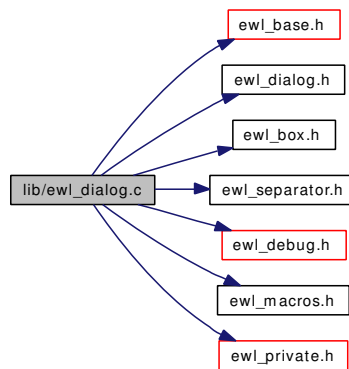
8.61.4.1 Ewl_Config_Cache ewl_config_cache

system debug data

8.62 lib/ewl_dialog.c File Reference

```
#include "ewl_base.h"
#include "ewl_dialog.h"
#include "ewl_box.h"
#include "ewl_separator.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_dialog.c:



Functions

- **Ewl_Position ewl_dialog_action_position_get (Ewl_Dialog *d)**
Checks the action area position for a dialog.
- **void ewl_dialog_action_position_set (Ewl_Dialog *d, Ewl_Position pos)**
Changes the action area position for a dialog.
- **Ewl_Position ewl_dialog_active_area_get (Ewl_Dialog *d)**
Retrieve the location of the dialog widgets will be packed into.
- **void ewl_dialog_active_area_set (Ewl_Dialog *d, Ewl_Position pos)**
Sets the section of the dialog that will have widgets packed into it.
- **unsigned int ewl_dialog_has_separator_get (Ewl_Dialog *dialog)**
Checks if dialog has a separator or not.
- **void ewl_dialog_has_separator_set (Ewl_Dialog *dialog, unsigned int has_sep)**
Sets the separator of dialog.
- **int ewl_dialog_init (Ewl_Dialog *dialog)**
Initialize an internal dialog to starting values.

- **Ewl_Widget * ewl_dialog_new** (void)

Create a new internal dialog.

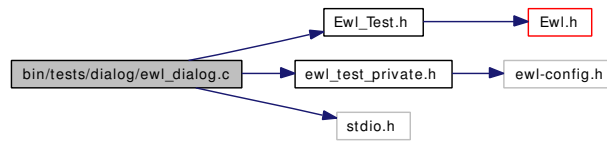
8.63 bin/tests/dialog/ewl_dialog.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_dialog.c:



Functions

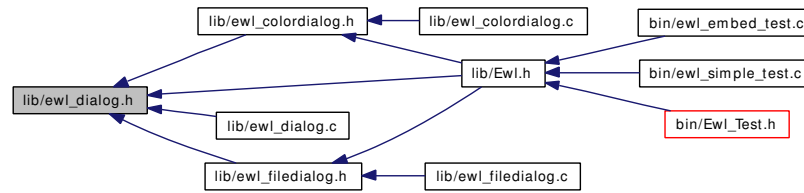
- `void test_info (Ewl_Test *test)`

8.63.1 Function Documentation

8.63.1.1 `void test_info (Ewl_Test * test)`

8.64 lib/ewl_dialog.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Dialog**

*Inherits from **Ewl_Window** (p. 788) and extends to provide a dialog widget.*

Defines

- #define **EWL_DIALOG**(dialog) ((**Ewl_Dialog** *) dialog)
- #define **EWL_DIALOG_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_DIALOG_TYPE))
- #define **EWL_DIALOG_TYPE** "dialog"

Typedefs

- typedef **Ewl_Dialog** **Ewl_Dialog**

Functions

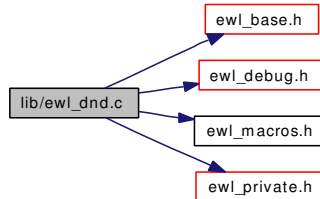
- **Ewl_Position** ewl_dialog_action_position_get (**Ewl_Dialog** *dialog)
Checks the action area position for a dialog.
- void ewl_dialog_action_position_set (**Ewl_Dialog** *d, **Ewl_Position** pos)
Changes the action area position for a dialog.
- **Ewl_Position** ewl_dialog_active_area_get (**Ewl_Dialog** *d)
Retrieve the location of the dialog widgets will be packed into.
- void ewl_dialog_active_area_set (**Ewl_Dialog** *d, **Ewl_Position** pos)
Sets the section of the dialog that will have widgets packed into it.
- unsigned int ewl_dialog_has_separator_get (**Ewl_Dialog** *dialog)
Checks if dialog has a separator or not.
- void ewl_dialog_has_separator_set (**Ewl_Dialog** *dialog, unsigned int has_sep)
Sets the separator of dialog.

- `int ewl_dialog_init (Ewl_Dialog *dialog)`
Initialize an internal dialog to starting values.
- `Ewl_Widget * ewl_dialog_new (void)`
Create a new internal dialog.

8.65 lib/ewl_dnd.c File Reference

```
#include "ewl_base.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_dnd.c:



Defines

- `#define EWL_DND_WINDOW_ROOT 0`

Functions

- `int ewl_dnd_accepted_types_contains (Ewl_Widget *w, char *type)`
: Verifies the specified widget accepts the given mimetype
- `const char ** ewl_dnd_accepted_types_get (Ewl_Widget *w)`
: Gets the mimetypes the designated widget can accept for DND
- `void ewl_dnd_accepted_types_set (Ewl_Widget *w, const char **types)`
: Sets the mimetypes the designated widget can accept for DND
- `void ewl_dnd_disable (void)`
Disables DND.
- `void ewl_dnd_drag_drop (Ewl_Widget *w)`
Tells the widget to start dragging.
- `void ewl_dnd_drag_start (Ewl_Widget *w)`
Tells the widget to start dragging.
- `void ewl_dnd_drag_widget_clear (void)`
Clears the current DND widget.
- `Ewl_Widget * ewl_dnd_drag_widget_get (void)`
Retrieves the current DND widget.
- `void ewl_dnd_enable (void)`

Enables DND.

- `int ewl_dnd_init (void)`
- `int ewl_dnd_provided_types_contains (Ewl_Widget *w, char *type)`
: *Verifies the specified widget provides the given mimetype*
- `char ** ewl_dnd_provided_types_get (Ewl_Widget *w)`
: *Gets the mimetypes the designated widget can provide for DND*
- `void ewl_dnd_provided_types_set (Ewl_Widget *w, const char **types)`
: *Sets the mimetypes the designated widget can provide for DND*
- `void ewl_dnd_shutdown (void)`
- `int ewl_dnd_status_get (void)`
Retrieves the current DND status.

Variables

- `int EWL_CALLBACK_DND_DATA_RECEIVED`
- `int EWL_CALLBACK_DND_DATA_REQUEST`
- `int EWL_CALLBACK_DND_DROP`
- `int EWL_CALLBACK_DND_ENTER`
- `int EWL_CALLBACK_DND_LEAVE`
- `int EWL_CALLBACK_DND_POSITION`

8.65.1 Define Documentation

8.65.1.1 `#define EWL_DND_WINDOW_ROOT 0`

8.66 lib/ewl_dnd.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- **int ewl_dnd_accepted_types_contains** (Ewl_Widget *w, char *type)
: *Verifies the specified widget accepts the given mimetype*
- **const char ** ewl_dnd_accepted_types_get** (Ewl_Widget *w)
: *Gets the mimetypes the designated widget can accept for DND*
- **void ewl_dnd_accepted_types_set** (Ewl_Widget *w, const char **types)
: *Sets the mimetypes the designated widget can accept for DND*
- **void ewl_dnd_disable** (void)
: *Disables DND.*
- **void ewl_dnd_drag_drop** (Ewl_Widget *w)
: *Tells the widget to start dragging.*
- **void ewl_dnd_drag_start** (Ewl_Widget *w)
: *Tells the widget to start dragging.*
- **void ewl_dnd_drag_widget_clear** (void)
: *Clears the current DND widget.*
- **Ewl_Widget * ewl_dnd_drag_widget_get** (void)
: *Retrieves the current DND widget.*
- **void ewl_dnd_enable** (void)
: *Enables DND.*
- **int ewl_dnd_init** (void)
- **int ewl_dnd_provided_types_contains** (Ewl_Widget *w, char *type)
: *Verifies the specified widget provides the given mimetype*
- **char ** ewl_dnd_provided_types_get** (Ewl_Widget *w)
: *Gets the mimetypes the designated widget can provide for DND*
- **void ewl_dnd_provided_types_set** (Ewl_Widget *w, const char **types)
: *Sets the mimetypes the designated widget can provide for DND*
- **void ewl_dnd_shutdown** (void)
- **int ewl_dnd_status_get** (void)
: *Retrieves the current DND status.*

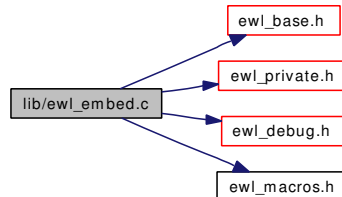
Variables

- int EWL_CALLBACK_DND_DATA_RECEIVED
- int EWL_CALLBACK_DND_DATA_REQUEST
- int EWL_CALLBACK_DND_DROP
- int EWL_CALLBACK_DND_ENTER
- int EWL_CALLBACK_DND_LEAVE
- int EWL_CALLBACK_DND_POSITION

8.67 lib/ewl_embed.c File Reference

```
#include "ewl_base.h"
#include "ewl_private.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
```

Include dependency graph for ewl_embed.c:



Functions

- **Ewl_Embed * ewl_embed_active_embed_get** (void)
Returns the currently active embed widget.
- **void ewl_embed_active_set** (Ewl_Embed *embed, unsigned int act)
Sets if the given embed `embed` is active or not.
- **void ewl_embed_cache_cleanup** (Ewl_Embed *emb)
This will destroy the object cache associated with the embed object.
- **void * ewl_embed_canvas_set** (Ewl_Embed *emb, void *canvas, Ewl_Embed_Window *canvas_window)
Change the canvas used by the embedded container.
- **Ewl_Embed * ewl_embed_canvas_window_find** (Ewl_Embed_Window *window)
Find an ewl embed by its evas window.
- **void ewl_embed_cb_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- **void ewl_embed_cb_destroy** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- **void ewl_embed_cb_focus_out** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- **void ewl_embed_cb_realize** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- **void ewl_embed_cb_unrealize** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- **void ewl_embed_coord_to_screen** (Ewl_Embed *e, int xx, int yy, int *x, int *y)
Maps coordinates from the Evas to screen coordinates.
- **void ewl_embed_desktop_size_get** (Ewl_Embed *e, int *w, int *h)

Retrieve the size of the root window.

- void **ewl_embed_dnd_aware_remove** (**Ewl_Embed** *embed)
Cancels an embed as being DND aware.
- void **ewl_embed_dnd_aware_set** (**Ewl_Embed** *embed)
Set an embed as being DND aware.
- void **ewl_embed_dnd_data_received_feed** (**Ewl_Embed** *embed, char *type, void *data, unsigned int len, unsigned int format)
Sends the event for selection data received into an embed.
- void **ewl_embed_dnd_data_request_feed** (**Ewl_Embed** *embed, void *handle, char *type)
Sends the request event for selection data received into an embed.
- const char * **ewl_embed_dnd_drop_feed** (**Ewl_Embed** *embed, int x, int y, int internal)
Sends the event for a DND drop into an embed.
- const char * **ewl_embed_dnd_position_feed** (**Ewl_Embed** *embed, int x, int y, int *px, int *py, int *pw, int *ph)
Sends the event for a DND position into an embed.
- const char * **ewl_embed_engine_name_get** (**Ewl_Embed** *embed)
Get the current engine on an embed.
- int **ewl_embed_engine_name_set** (**Ewl_Embed** *embed, const char *engine)
Changes the current engine on an embed to the specified engine.
- int **ewl_embed_focus_get** (**Ewl_Embed** *embed)
Retrieve the boolean value that indicates if the embed takes focus.
- void **ewl_embed_focus_set** (**Ewl_Embed** *embed, int f)
Sets the boolean flag in the embed to determine if it takes focus.
- **Ewl_Widget** * **ewl_embed_focused_widget_get** (**Ewl_Embed** *embed)
Get the current focused widget in the embed.
- void **ewl_embed_focused_widget_set** (**Ewl_Embed** *embed, **Ewl_Widget** *w)
Set the current focused widget in the embed.
- void **ewl_embed_font_path_add** (char *path)
Add a font path to all embeds after realized.
- void **ewl_embed_freeze** (**Ewl_Embed** *e)
Freeze redraws and events on the specified embed.
- void **ewl_embed_info_widgets_cleanup** (**Ewl_Embed** *e, **Ewl_Widget** *w)
This will check to see if the given widget is one of the last selected, clicked, mouse_in or the drag-n-drop widget and if so, set them to NULL.

- **int ewl_embed_init (Ewl_Embed *w)**
initialize a embed to default values and callbacks
- **void ewl_embed_key_down_feed (Ewl_Embed *embed, const char *keyname, unsigned int mods)**
Sends the event for a key press into an embed.
- **void ewl_embed_key_up_feed (Ewl_Embed *embed, const char *keyname, unsigned int mods)**
Sends the event for a key release into an embed.
- **void ewl_embed_mouse_cursor_set (Ewl_Widget *w)**
This will retrieve the attached cursor off of the given widget and display it. If there is no widget on the widget it will use the default cursor.
- **void ewl_embed_mouse_down_feed (Ewl_Embed *embed, int b, int clicks, int x, int y, unsigned int mods)**
Sends the event for a mouse button press into an embed.
- **void ewl_embed_mouse_move_feed (Ewl_Embed *embed, int x, int y, unsigned int mods)**
Sends the event for a mouse button release into an embed.
- **void ewl_embed_mouse_out_feed (Ewl_Embed *embed, int x, int y, unsigned int mods)**
Sends a mouse out event to the last focused widget.
- **void ewl_embed_mouse_up_feed (Ewl_Embed *embed, int b, int x, int y, unsigned int mods)**
Sends the event for a mouse button release into an embed.
- **void ewl_embed_mouse_wheel_feed (Ewl_Embed *embed, int x, int y, int z, int dir, unsigned int mods)**
Sends a mouse out event to the last focused widget.
- **Ewl_Widget * ewl_embed_new (void)**
Allocate and initialize a new embed.
- **void ewl_embed_object_cache (Ewl_Embed *e, void *obj)**
Caches the specified object for later reuse.
- **void * ewl_embed_object_request (Ewl_Embed *e, char *type)**
Retrieves an object of type type from the embed cache, or NULL if none found.
- **void ewl_embed_position_get (Ewl_Embed *e, int *x, int *y)**
Retrieve the position of the embed.
- **void ewl_embed_shutdown (void)**
- **void ewl_embed_tab_order_append (Ewl_Embed *e, Ewl_Widget *w)**

Moves the widget w to the end of the tab order.

- void **ewl_embed_tab_order_insert** (**Ewl_Embed** *e, **Ewl_Widget** *w, unsigned int idx)

Moves the given widget w to the position idx.

- void **ewl_embed_tab_order_insert_after** (**Ewl_Embed** *e, **Ewl_Widget** *w, **Ewl_Widget** *after)

Inserts the w widget into the tab order after the after widget.

- void **ewl_embed_tab_order_insert_before** (**Ewl_Embed** *e, **Ewl_Widget** *w, **Ewl_Widget** *before)

Inserts the w widget into the tab order before the before widget.

- void **ewl_embed_tab_order_next** (**Ewl_Embed** *e)

Changes focus to the next widget in the circular tab order list.

- void **ewl_embed_tab_order_prepend** (**Ewl_Embed** *e, **Ewl_Widget** *w)

Moves the widget w to the front of the tab order list.

- void **ewl_embed_tab_order_previous** (**Ewl_Embed** *e)

Changes focus to the next widget in the circular tab order list.

- void **ewl_embed_tab_order_remove** (**Ewl_Embed** *e, **Ewl_Widget** *w)

Removes the widget w from the tab order list for e.

- void **ewl_embed_thaw** (**Ewl_Embed** *e)

Thaw redraws and events on the specified embed.

- **Ewl_Embed** * **ewl_embed_widget_find** (**Ewl_Widget** *w)

Find an ewl embed by a widget inside.

- void **ewl_embed_window_position_get** (**Ewl_Embed** *e, int *x, int *y)

Retrieve the position of the window.

Variables

- **Ecore_List** * **ewl_embed_list** = NULL

8.67.1 Function Documentation

- 8.67.1.1 void ewl_embed_cb_configure (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.67.1.2 void ewl_embed_cb_destroy (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.67.1.3 void ewl_embed_cb_focus_out (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.67.1.4 void ewl_embed_cb_realize (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.67.1.5 void ewl_embed_cb_unrealize (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)

8.67.2 Variable Documentation

- 8.67.2.1 Ecore_List* ewl_embed_list = NULL

8.68 lib/ewl_embed.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Embed**

*Inherits from **Ewl_Overlay** (p. 712) that acts as a top level widget for interacting with the canvas.*

Defines

- `#define EWL_EMBED(widget) ((Ewl_Embed *) widget)`
*Typecast a pointer to an **Ewl_Embed** (p. 629) pointer.*
- `#define EWL_EMBED_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_EMBED_TYPE))`
- `#define EWL_EMBED_TYPE "embed"`
- `#define EWL_EMBED_WINDOW(window) ((Ewl_Embed_Window *)window)`
Typecast a pointer to an canvas window.

Typedefs

- `typedef Ewl_Embed Ewl_Embed`
- `typedef void * Ewl_Embed_Window`

Functions

- `Ewl_Embed * ewl_embed_active_embed_get (void)`
Returns the currently active embed widget.
- `void ewl_embed_active_set (Ewl_Embed *embed, unsigned int act)`
Sets if the given embed `embed` is active or not.
- `void ewl_embed_cache_cleanup (Ewl_Embed *emb)`
This will destroy the object cache associated with the embed object.
- `void * ewl_embed_canvas_set (Ewl_Embed *emb, void *canvas, Ewl_Embed_Window *canvas_window)`
Change the canvas used by the embedded container.
- `Ewl_Embed * ewl_embed_canvas_window_find (Ewl_Embed_Window *win)`

Find an ewl embed by its evas window.

- void **ewl_embed_cb_configure** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_embed_cb_destroy** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_embed_cb_focus_out** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_embed_cb_realize** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_embed_cb_unrealize** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_embed_coord_to_screen** (Ewl_Embed *e, int xx, int yy, int *x, int *y)

Maps coordinates from the Evas to screen coordinates.

- void **ewl_embed_desktop_size_get** (Ewl_Embed *e, int *w, int *y)

Retrieve the size of the root window.

- void **ewl_embed_dnd_aware_remove** (Ewl_Embed *embed)

Cancels an embed as being DND aware.

- void **ewl_embed_dnd_aware_set** (Ewl_Embed *embed)

Set an embed as being DND aware.

- void **ewl_embed_dnd_data_received_feed** (Ewl_Embed *embed, char *type, void *data, unsigned int len, unsigned int format)

Sends the event for selection data received into an embed.

- void **ewl_embed_dnd_data_request_feed** (Ewl_Embed *embed, void *handle, char *type)

Sends the request event for selection data received into an embed.

- const char * **ewl_embed_dnd_drop_feed** (Ewl_Embed *embed, int x, int y, int internal)

Sends the event for a DND drop into an embed.

- const char * **ewl_embed_dnd_position_feed** (Ewl_Embed *embed, int x, int y, int *, int *, int *, int *)

Sends the event for a DND position into an embed.

- const char * **ewl_embed_engine_name_get** (Ewl_Embed *embed)

Get the current engine on an embed.

- int **ewl_embed_engine_name_set** (Ewl_Embed *embed, const char *engine)

Changes the current engine on an embed to the specified engine.

- int **ewl_embed_focus_get** (Ewl_Embed *embed)

Retrieve the boolean value that indicates if the embed takes focus.

- void **ewl_embed_focus_set** (Ewl_Embed *embed, int f)

Sets the boolean flag in the embed to determine if it takes focus.

- Ewl_Widget * **ewl_embed_focused_widget_get** (Ewl_Embed *e)

Get the current focused widget in the embed.

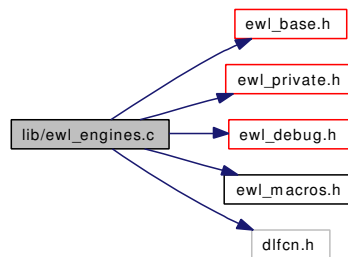
- void **ewl_embed_focused_widget_set** (**Ewl_Embed** *e, **Ewl_Widget** *w)
Set the current focused widget in the embed.
- void **ewl_embed_font_path_add** (char *path)
Add a font path to all embeds after realized.
- void **ewl_embed_freeze** (**Ewl_Embed** *e)
Freeze redraws and events on the specified embed.
- void **ewl_embed_info_widgets_cleanup** (**Ewl_Embed** *e, **Ewl_Widget** *w)
This will check to see if the given widget is one of the last selected, clicked, mouse_in or the drag-n-drop widget and if so, set them to NULL.
- int **ewl_embed_init** (**Ewl_Embed** *emb)
initialize a embed to default values and callbacks
- void **ewl_embed_key_down_feed** (**Ewl_Embed** *embed, const char *keyname, unsigned int modifiers)
Sends the event for a key press into an embed.
- void **ewl_embed_key_up_feed** (**Ewl_Embed** *embed, const char *keyname, unsigned int modifiers)
Sends the event for a key release into an embed.
- void **ewl_embed_mouse_cursor_set** (**Ewl_Widget** *w)
This will retrieve the attached cursor off of the given widget and display it. If there is no widget on the widget it will use the default cursor.
- void **ewl_embed_mouse_down_feed** (**Ewl_Embed** *embed, int b, int clicks, int x, int y, unsigned int modifiers)
Sends the event for a mouse button press into an embed.
- void **ewl_embed_mouse_move_feed** (**Ewl_Embed** *embed, int x, int y, unsigned int modifiers)
Sends the event for a mouse button release into an embed.
- void **ewl_embed_mouse_out_feed** (**Ewl_Embed** *embed, int x, int y, unsigned int modifiers)
Sends a mouse out event to the last focused widget.
- void **ewl_embed_mouse_up_feed** (**Ewl_Embed** *embed, int b, int x, int y, unsigned int modifiers)
Sends the event for a mouse button release into an embed.
- void **ewl_embed_mouse_wheel_feed** (**Ewl_Embed** *embed, int x, int y, int z, int dir, unsigned int mods)
Sends a mouse out event to the last focused widget.
- **Ewl_Widget** * **ewl_embed_new** (void)
Allocate and initialize a new embed.

- void **ewl_embed_object_cache** (**Ewl_Embed** *e, void *obj)
Caches the specified object for later reuse.
- void * **ewl_embed_object_request** (**Ewl_Embed** *e, char *type)
Retrieves an object of type type from the embed cache, or NULL if none found.
- void **ewl_embed_position_get** (**Ewl_Embed** *e, int *x, int *y)
Retrieve the position of the embed.
- void **ewl_embed_shutdown** (void)
- void **ewl_embed_tab_order_append** (**Ewl_Embed** *e, **Ewl_Widget** *w)
Moves the widget w to the end of the tab order.
- void **ewl_embed_tab_order_insert** (**Ewl_Embed** *e, **Ewl_Widget** *w, unsigned int idx)
Moves the given widget w to the position idx.
- void **ewl_embed_tab_order_insert_after** (**Ewl_Embed** *e, **Ewl_Widget** *w, **Ewl_Widget** *after)
Inserts the w widget into the tab order after the after widget.
- void **ewl_embed_tab_order_insert_before** (**Ewl_Embed** *e, **Ewl_Widget** *w, **Ewl_Widget** *before)
Inserts the w widget into the tab order before the before widget.
- void **ewl_embed_tab_order_next** (**Ewl_Embed** *e)
Changes focus to the next widget in the circular tab order list.
- void **ewl_embed_tab_order_prepend** (**Ewl_Embed** *e, **Ewl_Widget** *w)
Moves the widget w to the front of the tab order list.
- void **ewl_embed_tab_order_previous** (**Ewl_Embed** *e)
Changes focus to the next widget in the circular tab order list.
- void **ewl_embed_tab_order_remove** (**Ewl_Embed** *e, **Ewl_Widget** *w)
Removes the widget w from the tab order list for e.
- void **ewl_embed_thaw** (**Ewl_Embed** *e)
Thaw redraws and events on the specified embed.
- **Ewl_Embed** * **ewl_embed_widget_find** (**Ewl_Widget** *w)
Find an ewl embed by a widget inside.
- void **ewl_embed_window_position_get** (**Ewl_Embed** *e, int *x, int *y)
Retrieve the position of the window.

8.69 lib/ewl_engines.c File Reference

```
#include "ewl_base.h"
#include "ewl_private.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include <dlfcn.h>
```

Include dependency graph for ewl_engines.c:



Defines

- `#define EWL_ENGINE_DIR "engines"`

Typedefs

- `typedef enum Ewl_Engine_Hook_Type Ewl_Engine_Hook_Type`

Enumerations

- `enum Ewl_Engine_Hook_Type { EWL_ENGINE_HOOK_TYPE_WINDOW, EWL_ENGINE_HOOK_TYPE_CANVAS, EWL_ENGINE_HOOK_TYPE_THEME, EWL_ENGINE_HOOK_TYPE_POINTER }`

Functions

- `void ewl_engine_canvas_damage_add (Ewl_Embed *embed, int x, int y, int w, int h)`
- `void ewl_engine_canvas_freeze (Ewl_Embed *embed)`
Freezes the canvas.
- `void ewl_engine_canvas_output_set (Ewl_Embed *emb, int x, int y, int width, int height)`
Sets up the canvas.
- `void ewl_engine_canvas_render (Ewl_Embed *embed)`
Renders the canvas.

- void **ewl_engine_canvas_setup** (Ewl_Window *win, int debug)
Sets up the canvas.
- void **ewl_engine_canvas_thaw** (Ewl_Embed *embed)
Thaws the canvas.
- void **ewl_engine_desktop_size_get** (Ewl_Embed *embed, int *w, int *h)
Sets the dnd awareness.
- void **ewl_engine_embed_dnd_aware_set** (Ewl_Embed *embed)
Sets the dnd awareness.
- int **ewl_engine_embed_dnd_drag_data_send** (Ewl_Embed *embed, void *handle, void *data, int size)
Sends dnd data to the drop recipient.
- void **ewl_engine_embed_dnd_drag_drop** (Ewl_Embed *embed)
Completes the dnd process on an embed.
- void **ewl_engine_embed_dnd_drag_start** (Ewl_Embed *embed)
Starts the dnd process on an embed.
- void **ewl_engine_embed_dnd_drag_types_set** (Ewl_Embed *embed, char **types, unsigned int num)
Sets the dnd types provided by this embed.
- int **ewl_engine_keyboard_grab** (Ewl_Window *win)
Sets the keyboard grab.
- void **ewl_engine_keyboard_ungrab** (Ewl_Window *win)
Sets the keyboard ungrab.
- Ecore_List * **ewl_engine_names_get** (void)
Retrieves a list of available engine names.
- Ewl_Engine * **ewl_engine_new** (const char *name, int *argc, char **argv)
*Retrieves, or creates the given **Ewl_Engine** (p. 633), or NULL on failure.*
- int **ewl_engine_pointer_data_new** (Ewl_Embed *embed, unsigned int *data, int w, int h)
Creates a new pointer from ARGB data.
- void **ewl_engine_pointer_free** (Ewl_Embed *embed, int pointer)
Free the identified pointer.
- int **ewl_engine_pointer_get** (Ewl_Embed *embed)
Retrieve the currently used pointer.
- int **ewl_engine_pointer_grab** (Ewl_Window *win)
Sets the pointer grab.

- void **ewl_engine_pointer_set** (**Ewl_Embed** *embed, int pointer)
Change the current pointer to the one identified.
- void **ewl_engine_pointer_ungrab** (**Ewl_Window** *win)
Sets the pointer ungrab.
- void **ewl_engine_theme_freeze** (**Ewl_Embed** *embed)
Freeze the theme system.
- void * **ewl_engine_theme_object_add** (**Ewl_Embed** *embed)
Add a theme object.
- void **ewl_engine_theme_object_clip_set** (**Ewl_Embed** *embed, void *obj, void *clip)
Clip set a theme object.
- void **ewl_engine_theme_object_clip_unset** (**Ewl_Embed** *embed, void *obj)
Clip unset a theme object.
- void **ewl_engine_theme_object_del** (**Ewl_Embed** *embed, void *obj)
Del a theme object.
- void **ewl_engine_theme_object_file_set** (**Ewl_Embed** *embed, void *obj, char *file, char *group)
File set a theme object.
- void **ewl_engine_theme_object_hide** (**Ewl_Embed** *embed, void *obj)
Hide a theme object.
- void **ewl_engine_theme_object_move** (**Ewl_Embed** *embed, void *obj, int x, int y)
Move a theme object.
- void **ewl_engine_theme_object_resize** (**Ewl_Embed** *embed, void *obj, int width, int height)
Resize a theme object.
- void **ewl_engine_theme_object_show** (**Ewl_Embed** *embed, void *obj)
Show a theme object.
- void **ewl_engine_theme_thaw** (**Ewl_Embed** *embed)
Thaw the theme system.
- void * **ewl_engine_theme_widget_group** (**Ewl_Widget** *w)
Create a grouping for theme objects.
- void **ewl_engine_window_borderless_set** (**Ewl_Window** *win)
Sets the window borderless.

- void **ewl_engine_window_destroy** (Ewl_Window *win)
Destroys the window.
- void **ewl_engine_window_dialog_set** (Ewl_Window *win)
Sets the window as a dialog.
- void **ewl_engine_window_geometry_get** (Ewl_Window *win, int root, int *width, int *height)
Retrieves the window geometry.
- void **ewl_engine_window_hide** (Ewl_Window *win)
Hides the window.
- void **ewl_engine_window_lower** (Ewl_Window *win)
Lowers the window.
- void **ewl_engine_window_min_max_size_set** (Ewl_Window *win)
Sets the window min/max size.
- void **ewl_engine_window_move** (Ewl_Window *win)
Moves the window.
- void **ewl_engine_window_name_class_set** (Ewl_Window *win)
Sets the name and class of the window.
- void **ewl_engine_window_new** (Ewl_Window *win)
Creates a new window.
- void **ewl_engine_window_raise** (Ewl_Window *win)
Raises the window.
- void **ewl_engine_window_resize** (Ewl_Window *win)
Resizes the window.
- void **ewl_engine_window_selection_text_set** (Ewl_Window *win, const char *txt)
Sets the window selection text.
- void **ewl_engine_window_show** (Ewl_Window *win)
Shows the window.
- void **ewl_engine_window_states_set** (Ewl_Window *win)
Sets the window state values.
- void **ewl_engine_window_title_set** (Ewl_Window *win)
Sets the title of the window.
- void **ewl_engine_window_transient_for** (Ewl_Window *win)
Sets the window transient.

- `int ewl_engines_init (void)`
Initialises the engine system.
- `void ewl_engines_shutdown (void)`
Shutdown the engine system.

8.69.1 Define Documentation

8.69.1.1 `#define EWL_ENGINE_DIR "engines"`

8.69.2 Typedef Documentation

8.69.2.1 `typedef enum Ewl_Engine_Hook_Type Ewl_Engine_Hook_Type`

8.69.3 Enumeration Type Documentation

8.69.3.1 `enum Ewl_Engine_Hook_Type`

Enumerator:

EWL_ENGINE_HOOK_TYPE_WINDOW
EWL_ENGINE_HOOK_TYPE_CANVAS
EWL_ENGINE_HOOK_TYPE_THEME
EWL_ENGINE_HOOK_TYPE_POINTER

8.69.4 Function Documentation

8.69.4.1 `void ewl_engine_canvas_damage_add (Ewl_Embed * embed, int x, int y, int w, int h)`

8.69.4.2 `void ewl_engine_canvas_freeze (Ewl_Embed * embed)`

Freezes the canvas.

Parameters:

embed,: the embed to work with

Returns:

Returns no value

8.69.4.3 `void ewl_engine_canvas_output_set (Ewl_Embed * emb, int x, int y, int width, int height)`

Sets up the canvas.

Parameters:

emb,: The embed to work with

x,: The x coord to set

y,: The y coord to set

width,: The width to set

height,: The height to set

Returns:

Returns no value

8.69.4.4 void ewl_engine_canvas_render (Ewl_Embed * *embed*)

Renders the canvas.

Parameters:

embed,: the embed to work with

Returns:

Returns no value

8.69.4.5 void ewl_engine_canvas_setup (Ewl_Window * *win*, int *debug*)

Sets up the canvas.

Parameters:

win,: the window to work with

debug,: The debug setting

Returns:

Returns no value

8.69.4.6 void ewl_engine_canvas_thaw (Ewl_Embed * *embed*)

Thaws the canvas.

Parameters:

embed,: the embed to work with

Returns:

Returns no value

8.69.4.7 void ewl_engine_desktop_size_get (Ewl_Embed * *embed*, int * *w*, int * *h*)

Sets the dnd awareness.

Parameters:

embed,: the embed to work with
w,: a pointer where the width of the desktop will be stored
h,: a pointer where the height pf the desktop will be stored

Returns:

Returns no value

8.69.4.8 void ewl_engine_embed_dnd_aware_set (Ewl_Embed * *embed*)

Sets the dnd awareness.

Parameters:

embed,: the embed to work with

Returns:

Returns no value

8.69.4.9 int ewl_engine_embed_dnd_drag_data_send (Ewl_Embed * *embed*, void * *handle*, void * *data*, int *size*)

Sends dnd data to the drop recipient.

Parameters:

embed,: the embed to work with
handle,: The DND handle to work with
data,: The data to send
size,: The size of the data being sent

Returns:

Returns no value

8.69.4.10 void ewl_engine_embed_dnd_drag_drop (Ewl_Embed * *embed*)

Completes the dnd process on an embed.

Parameters:

embed,: the embed to work with

Returns:

Returns no value

8.69.4.11 void ewl_engine_embed_dnd_drag_start (Ewl_Embed * *embed*)

Starts the dnd process on an embed.

Parameters:

embed,: the embed to work with

Returns:

Returns no value

**8.69.4.12 void ewl_engine_embed_dnd_drag_types_set (Ewl_Embed * *embed*,
char ** *types*, unsigned int *num*)**

Sets the dnd types provided by this embed.

Parameters:

embed,: the embed to work with

types,: The dnd types to set

num,: The number of types

Returns:

Returns no value

8.69.4.13 int ewl_engine_keyboard_grab (Ewl_Window * *win*)

Sets the keyboard grab.

Parameters:

win,: the window to work with

Returns:

The keyboard grab value

8.69.4.14 void ewl_engine_keyboard_ungrab (Ewl_Window * *win*)

Sets the keyboard ungrab.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.15 `Ecore_List* ewl_engine_names_get (void)`

Retrieves a list of available engine names.

Returns:

Returns a list of engine names

8.69.4.16 `Ewl_Engine* ewl_engine_new (const char * name, int * argc, char ** argv)`

Retrieves, or creates the given **Ewl_Engine** (p. 633), or NULL on failure.

Parameters:

name,: The name of the engine to create

argc,: Arguments count

argv,: Arguments

Returns:

Returns the **Ewl_Engine** (p. 633) or NULL on failure

8.69.4.17 `int ewl_engine_pointer_data_new (Ewl_Engine * embed, unsigned int * data, int w, int h)`

Creates a new pointer from ARGB data.

Returns:

Returns a pointer id on success, zero on failure.

8.69.4.18 `void ewl_engine_pointer_free (Ewl_Engine * embed, int pointer)`

Free the identified pointer.

Returns:

Returns no value.

8.69.4.19 `int ewl_engine_pointer_get (Ewl_Engine * embed)`

Retrieve the currently used pointer.

Returns:

Returns a pointer id on success, zero on failure.

8.69.4.20 int ewl_engine_pointer_grab (Ewl_Window * *win*)

Sets the pointer grab.

Parameters:

win,: the window to work with

Returns:

Returns the pointer grab value

8.69.4.21 void ewl_engine_pointer_set (Ewl_Embed * *embed*, int *pointer*)

Change the current pointer to the one identified.

Returns:

Returns no value.

8.69.4.22 void ewl_engine_pointer_ungrab (Ewl_Window * *win*)

Sets the pointer ungrab.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.23 void ewl_engine_theme_freeze (Ewl_Embed * *embed*)

Freeze the theme system.

Parameters:

embed,: Embed used to lookup the current theme engine.

Returns:

Returns no value

8.69.4.24 void* ewl_engine_theme_object_add (Ewl_Embed * *embed*)

Add a theme object.

Returns:

Returns a new theme object on success, NULL on failure

8.69.4.25 `void ewl_engine_theme_object_clip_set (Ewl_Embed * embed, void * obj, void * clip)`

Clip set a theme object.

Returns:

Returns no value.

8.69.4.26 `void ewl_engine_theme_object_clip_unset (Ewl_Embed * embed, void * obj)`

Clip unset a theme object.

Returns:

Returns no value.

8.69.4.27 `void ewl_engine_theme_object_del (Ewl_Embed * embed, void * obj)`

Del a theme object.

Returns:

Returns a new theme object on success, NULL on failure

8.69.4.28 `void ewl_engine_theme_object_file_set (Ewl_Embed * embed, void * obj, char * file, char * group)`

File set a theme object.

Returns:

Returns no value.

8.69.4.29 `void ewl_engine_theme_object_hide (Ewl_Embed * embed, void * obj)`

Hide a theme object.

Returns:

Returns no value.

8.69.4.30 `void ewl_engine_theme_object_move (Ewl_Embed * embed, void * obj, int x, int y)`

Move a theme object.

Returns:

Returns no value.

8.69.4.31 void ewl_engine_theme_object_resize (Ewl_Embed * *embed*, void * *obj*, int *width*, int *height*)

Resize a theme object.

Returns:

Returns no value.

8.69.4.32 void ewl_engine_theme_object_show (Ewl_Embed * *embed*, void * *obj*)

Show a theme object.

Returns:

Returns no value.

8.69.4.33 void ewl_engine_theme_thaw (Ewl_Embed * *embed*)

Thaw the theme system.

Parameters:

embed,: Embed used to lookup the current theme engine.

Returns:

Returns no value

8.69.4.34 void* ewl_engine_theme_widget_group (Ewl_Widget * *w*)

Create a grouping for theme objects.

Returns:

Returns a new object group on success, NULL on failure

8.69.4.35 void ewl_engine_window_borderless_set (Ewl_Window * *win*)

Sets the window borderless.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.36 void ewl_engine_window_destroy (Ewl_Window * *win*)

Destroys the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.37 void ewl_engine_window_dialog_set (Ewl_Window * *win*)

Sets the window as a dialog.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.38 void ewl_engine_window_geometry_get (Ewl_Window * *win*, int *root*, int * *width*, int * *height*)

Retrieves the window geometry.

Parameters:

win,: the window to work with

root,: The root window to get the height for

width,: Where to store the width

height,: Where to store the height

Returns:

Returns no value

We dont' check win == null in case their querying after the root window

8.69.4.39 void ewl_engine_window_hide (Ewl_Window * *win*)

Hides the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.40 void ewl_engine_window_lower (Ewl_Window * *win*)

Lowers the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.41 void ewl_engine_window_min_max_size_set (Ewl_Window * *win*)

Sets the window min/max size.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.42 void ewl_engine_window_move (Ewl_Window * *win*)

Moves the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.43 void ewl_engine_window_name_class_set (Ewl_Window * *win*)

Sets the name and class of the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.44 void ewl_engine_window_new (Ewl_Window * *win*)

Creates a new window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.45 void ewl_engine_window_raise (Ewl_Window * *win*)

Raises the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.46 void ewl_engine_window_resize (Ewl_Window * *win*)

Resizes the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

**8.69.4.47 void ewl_engine_window_selection_text_set (Ewl_Window * *win*,
const char * *txt*)**

Sets the window selection text.

Parameters:

win,: the window to work with

txt,: The text to set as the selection

Returns:

Returns no value

8.69.4.48 void ewl_engine_window_show (Ewl_Window * *win*)

Shows the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.49 void ewl_engine_window_states_set (Ewl_Window * *win*)

Sets the window state values.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.50 void ewl_engine_window_title_set (Ewl_Window * *win*)

Sets the title of the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.51 void ewl_engine_window_transient_for (Ewl_Window * *win*)

Sets the window transient.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.69.4.52 int ewl_engines_init (void)

Initialises the engine system.

Returns:

Returns no value

8.69.4.53 void ewl_engines_shutdown (void)

Shutdown the engine system.

Returns:

Returns no value

8.70 lib/ewl_engines.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Engine**
Contains information on an Ewl engine.
- struct **Ewl_Engine_Info**
Contains the function pointers provided by an engine.

Defines

- #define **EWL_ENGINE(engine)** ((Ewl_Engine *)engine)

Typedefs

- typedef **Ewl_Engine** Ewl_Engine
- typedef enum **Ewl_Engine_Canvas_Hooks** Ewl_Engine_Canvas_Hooks
- typedef void(*) **Ewl_Engine_Cb_Canvas_Damage_Add** (Ewl_Embed *embed, int x, int y, int w, int h)
- typedef void(*) **Ewl_Engine_Cb_Canvas_Freeze** (Ewl_Embed *embed)
- typedef void(*) **Ewl_Engine_Cb_Canvas_Output_Set** (Ewl_Embed *emb, int x, int y, int width, int height)
- typedef void(*) **Ewl_Engine_Cb_Canvas_Render** (Ewl_Embed *embed)
- typedef void(*) **Ewl_Engine_Cb_Canvas_Setup** (Ewl_Window *win, int debug)
- typedef void(*) **Ewl_Engine_Cb_Canvas_Thaw** (Ewl_Embed *embed)
- typedef int(*) **Ewl_Engine_Cb_Keyboard_Grab** (Ewl_Window *win)
- typedef void(*) **Ewl_Engine_Cb_Keyboard_Ungrab** (Ewl_Window *win)
- typedef int(*) **Ewl_Engine_Cb_Pointer_Data_New** (Ewl_Embed *embed, unsigned int *data, int w, int h)
- typedef void(*) **Ewl_Engine_Cb_Pointer_Free** (Ewl_Embed *embed, int pointer)
- typedef int(*) **Ewl_Engine_Cb_Pointer_Get** (Ewl_Embed *embed)
- typedef int(*) **Ewl_Engine_Cb_Pointer_Grab** (Ewl_Window *win)
- typedef void(*) **Ewl_Engine_Cb_Pointer_Set** (Ewl_Embed *embed, int pointer)
- typedef void(*) **Ewl_Engine_Cb_Pointer_Ungrab** (Ewl_Window *win)
- typedef void(*) **Ewl_Engine_Cb_Theme_Clip_Add** (Ewl_Embed *embed)
- typedef void(*) **Ewl_Engine_Cb_Theme_Clip_Clipees_Get** (void *clip)
- typedef void(*) **Ewl_Engine_Cb_Theme_Clip_Color_Set** (void *clip, int r, int g, int b, int a)
- typedef void(*) **Ewl_Engine_Cb_Theme_Clip_Del** (void *clip)
- typedef char(*) **Ewl_Engine_Cb_Theme_Data_Get** (char *path, char *key)
- typedef void(*) **Ewl_Engine_Cb_Theme_Freeze** ()
- typedef void(*) **Ewl_Engine_Cb_Theme_Object_Add** (Ewl_Embed *embed)

- typedef void (*) **Ewl_Engine_Cb_Theme_Object_Clip_Set** (void *obj, void *clip)
- typedef void (*) **Ewl_Engine_Cb_Theme_Object_Clip_Unset** (void *obj)
- typedef void (*) **Ewl_Engine_Cb_Theme_Object_Color_Class_Set** (char *name, int r, int g, int b, int r2, int g2, int b2, int r3, int g3, int b3)
- typedef void (*) **Ewl_Engine_Cb_Theme_Object_Del** (void *obj)
- typedef void (*) **Ewl_Engine_Cb_Theme_Object_File_Set** (void *obj, char *path, char *group)
- typedef void (*) **Ewl_Engine_Cb_Theme_Object_Hide** (void *obj)
- typedef void (*) **Ewl_Engine_Cb_Theme_Object_Layer_Update** (**Ewl_Widget** *w)
- typedef void (*) **Ewl_Engine_Cb_Theme_Object_Load_Error** (void *obj)
- typedef void (*) **Ewl_Engine_Cb_Theme_Object_Max_Size_Get** (void *obj, int *w, int *h)
- typedef void (*) **Ewl_Engine_Cb_Theme_Object_Min_Size_Get** (void *obj, int *w, int *h)
- typedef void (*) **Ewl_Engine_Cb_Theme_Object_Move** (void *obj, int x, int y)
- typedef void (*) **Ewl_Engine_Cb_Theme_Object_Part_Text_Set** (void *obj, char *part, char *text)
- typedef void (*) **Ewl_Engine_Cb_Theme_Object_Resize** (void *obj, int w, int h)
- typedef void (*) **Ewl_Engine_Cb_Theme_Object_Show** (void *obj)
- typedef void (*) **Ewl_Engine_Cb_Theme_Object_Signal_Send** (void *obj, char *state, char *source)
- typedef void (*) **Ewl_Engine_Cb_Theme_Object_Stack_Add** (**Ewl_Widget** *w)
- typedef void (*) **Ewl_Engine_Cb_Theme_Thaw** ()
- typedef void (*) **Ewl_Engine_Cb_Theme_Widget_Group** (**Ewl_Widget** *w)
- typedef void (*) **Ewl_Engine_Cb_Window_Borderless_Set** (**Ewl_Window** *win)
- typedef void (*) **Ewl_Engine_Cb_Window_Desktop_Size_Get** (**Ewl_Embed** *embed, int *w, int *h)
- typedef void (*) **Ewl_Engine_Cb_Window_Destroy** (**Ewl_Window** *win)
- typedef void (*) **Ewl_Engine_Cb_Window_Dialog_Set** (**Ewl_Window** *win)
- typedef void (*) **Ewl_Engine_Cb_Window_Dnd_Aware_Set** (**Ewl_Embed** *embed)
- typedef int (*) **Ewl_Engine_Cb_Window_Dnd_Drag_Data_Send** (**Ewl_Embed** *embed, void *handle, void *data, int size)
- typedef void (*) **Ewl_Engine_Cb_Window_Dnd_Drag_Drop** (**Ewl_Embed** *embed)
- typedef void (*) **Ewl_Engine_Cb_Window_Dnd_Drag_Start** (**Ewl_Embed** *embed)
- typedef void (*) **Ewl_Engine_Cb_Window_Dnd_Drag_Types_Set** (**Ewl_Embed** *embed, char **types, unsigned int num)
- typedef void (*) **Ewl_Engine_Cb_Window_Geometry_Get** (**Ewl_Window** *win, int *width, int *height)
- typedef void (*) **Ewl_Engine_Cb_Window_Hide** (**Ewl_Window** *win)
- typedef void (*) **Ewl_Engine_Cb_Window_Lower** (**Ewl_Window** *win)
- typedef void (*) **Ewl_Engine_Cb_Window_Min_Max_Size_Set** (**Ewl_Window** *win)
- typedef void (*) **Ewl_Engine_Cb_Window_Move** (**Ewl_Window** *win)
- typedef void (*) **Ewl_Engine_Cb_Window_Name_Class_Set** (**Ewl_Window** *win)
- typedef void (*) **Ewl_Engine_Cb_Window_New** (**Ewl_Window** *win)
- typedef void (*) **Ewl_Engine_Cb_Window_Raise** (**Ewl_Window** *win)

- typedef void(*) Ewl_Engine_Cb_Window_Resize (Ewl_Window *win)
- typedef void(*) Ewl_Engine_Cb_Window_Selection_Text_Set (Ewl_Window *win, const char *txt)
- typedef void(*) Ewl_Engine_Cb_Window_Show (Ewl_Window *win)
- typedef void(*) Ewl_Engine_Cb_Window_States_Set (Ewl_Window *win)
- typedef void(*) Ewl_Engine_Cb_Window_Title_Set (Ewl_Window *win)
- typedef void(*) Ewl_Engine_Cb_Window_Transient_For (Ewl_Window *win)
- typedef Ewl_Engine_Info Ewl_Engine_Info
- typedef enum Ewl_Engine_Pointer_Hooks Ewl_Engine_Pointer_Hooks
- typedef enum Ewl_Engine_Theme_Hooks Ewl_Engine_Theme_Hooks
- typedef enum Ewl_Engine_Window_Hooks Ewl_Engine_Window_Hooks

Enumerations

- enum Ewl_Engine_Canvas_Hooks {
EWL_ENGINE_CANVAS_SETUP, EWL_ENGINE_CANVAS_OUTPUT_SET, EWL_ENGINE_CANVAS_RENDER, EWL_ENGINE_CANVAS_FREEZE,
EWL_ENGINE_CANVAS_THAW, EWL_ENGINE_CANVAS_DAMAGE_ADD, EWL_ENGINE_CANVAS_MAX }
- enum Ewl_Engine_Pointer_Hooks {
EWL_ENGINE_POINTER_DATA_NEW, EWL_ENGINE_POINTER_FREE, EWL_ENGINE_POINTER_GET, EWL_ENGINE_POINTER_SET, EWL_ENGINE_POINTER_MAX }
- enum Ewl_Engine_Theme_Hooks {
EWL_ENGINE_THEME_FREEZE, EWL_ENGINE_THEME_THAW, EWL_ENGINE_THEME_DATA_GET, EWL_ENGINE_THEME_WIDGET_GROUP,
EWL_ENGINE_THEME_OBJECT_ADD, EWL_ENGINE_THEME_OBJECT_DEL, EWL_ENGINE_THEME_OBJECT_MOVE, EWL_ENGINE_THEME_OBJECT_RESIZE,
EWL_ENGINE_THEME_OBJECT_SHOW, EWL_ENGINE_THEME_OBJECT_HIDE, EWL_ENGINE_THEME_OBJECT_CLIP_SET, EWL_ENGINE_THEME_OBJECT_CLIP_UNSET,
EWL_ENGINE_THEME_OBJECT_FILE_SET, EWL_ENGINE_THEME_OBJECT_LOAD_ERROR, EWL_ENGINE_THEME_OBJECT_MIN_SIZE_GET, EWL_ENGINE_THEME_OBJECT_MAX_SIZE_GET,
EWL_ENGINE_THEME_OBJECT_SIGNAL_SEND, EWL_ENGINE_THEME_OBJECT_PART_TEXT_SET, EWL_ENGINE_THEME_OBJECT_COLOR_CLASS_SET, EWL_ENGINE_THEME_CLIP_ADD,
EWL_ENGINE_THEME_CLIP_DEL, EWL_ENGINE_THEME_CLIP_CLIP_EES_GET, EWL_ENGINE_THEME_CLIP_COLOR_SET, EWL_ENGINE_THEME_WIDGET_STACK_ADD,
EWL_ENGINE_THEME_WIDGET_LAYER_UPDATE, EWL_ENGINE_THEME_MAX }
- enum Ewl_Engine_Window_Hooks {
EWL_ENGINE_WINDOW_NEW, EWL_ENGINE_WINDOW_DESTROY, EWL_ENGINE_WINDOW_MOVE, EWL_ENGINE_WINDOW_RESIZE,

```

EWL_ENGINE_WINDOW_MIN_MAX_SIZE_SET,    EWL_ENGINE_-
WINDOW_SHOW, EWL_ENGINE_WINDOW_HIDE, EWL_ENGINE_-
WINDOW_TITLE_SET,

EWL_ENGINE_WINDOW_NAME_CLASS_SET,      EWL_ENGINE_-
WINDOW_BORDERLESS_SET, EWL_ENGINE_WINDOW_DIALOG_-
SET, EWL_ENGINE_WINDOW_STATES_SET,

EWL_ENGINE_WINDOW_TRANSIENT_FOR,      EWL_ENGINE_-
WINDOW_RAISE,    EWL_ENGINE_WINDOW_LOWER,    EWL_-
ENGINE_WINDOW_KEYBOARD_GRAB,

EWL_ENGINE_WINDOW_KEYBOARD_UNGRAB,    EWL_ENGINE_-
WINDOW_POINTER_GRAB, EWL_ENGINE_WINDOW_POINTER_-
UNGRAB, EWL_ENGINE_WINDOW_SELECTION_TEXT_SET,

EWL_ENGINE_WINDOW_GEOMETRY_GET,      EWL_ENGINE_-
WINDOW_DND_AWARE_SET,      EWL_ENGINE_WINDOW_-
DESKTOP_SIZE_GET,    EWL_ENGINE_WINDOW_DND_DRAG_-
TYPES_SET,

EWL_ENGINE_WINDOW_DND_DRAG_START,    EWL_ENGINE_-
WINDOW_DND_DRAG_DROP,    EWL_ENGINE_WINDOW_DND_-
DRAG_DATA_SEND, EWL_ENGINE_WINDOW_MAX }

```

Functions

- void **ewl_engine_canvas_damage_add** (Ewl_Embed *embed, int x, int y, int w, int h)
- void **ewl_engine_canvas_freeze** (Ewl_Embed *embed)
Freezes the canvas.
- void **ewl_engine_canvas_output_set** (Ewl_Embed *emb, int x, int y, int width, int height)
Sets up the canvas.
- void **ewl_engine_canvas_render** (Ewl_Embed *embed)
Renders the canvas.
- void **ewl_engine_canvas_setup** (Ewl_Window *win, int debug)
Sets up the canvas.
- void **ewl_engine_canvas_thaw** (Ewl_Embed *embed)
Thaws the canvas.
- void **ewl_engine_desktop_size_get** (Ewl_Embed *embed, int *w, int *h)
Sets the dnd awareness.
- void **ewl_engine_embed_dnd_aware_set** (Ewl_Embed *embed)
Sets the dnd awareness.
- int **ewl_engine_embed_dnd_drag_data_send** (Ewl_Embed *embed, void *handle, void *data, int size)
Sends dnd data to the drop recipient.

- void **ewl_engine_embed_dnd_drag_drop** (Ewl_Embed *embed)
Completes the dnd process on an embed.
- void **ewl_engine_embed_dnd_drag_start** (Ewl_Embed *embed)
Starts the dnd process on an embed.
- void **ewl_engine_embed_dnd_drag_types_set** (Ewl_Embed *embed, char **types, unsigned int num)
Sets the dnd types provided by this embed.
- void **ewl_engine_engine_shutdown** (Ewl_Window *win)
- void **ewl_engine_event_handlers_init** (void)
- void **ewl_engine_event_handlers_shutdown** (void)
- int **ewl_engine_keyboard_grab** (Ewl_Window *win)
Sets the keyboard grab.
- void **ewl_engine_keyboard_ungrab** (Ewl_Window *win)
Sets the keyboard ungrab.
- Ecore_List * **ewl_engine_names_get** (void)
Retrieves a list of available engine names.
- **Ewl_Engine** * **ewl_engine_new** (const char *name, int *argc, char **argv)
*Retrieves, or creates the given **Ewl_Engine** (p. 633), or NULL on failure.*
- int **ewl_engine_pointer_data_new** (Ewl_Embed *embed, unsigned int *data, int w, int h)
Creates a new pointer from ARGB data.
- void **ewl_engine_pointer_free** (Ewl_Embed *embed, int pointer)
Free the identified pointer.
- int **ewl_engine_pointer_get** (Ewl_Embed *embed)
Retrieve the currently used pointer.
- int **ewl_engine_pointer_grab** (Ewl_Window *win)
Sets the pointer grab.
- void **ewl_engine_pointer_set** (Ewl_Embed *embed, int pointer)
Change the current pointer to the one identified.
- void **ewl_engine_pointer_ungrab** (Ewl_Window *win)
Sets the pointer ungrab.
- void **ewl_engine_window_borderless_set** (Ewl_Window *win)
Sets the window borderless.
- void **ewl_engine_window_destroy** (Ewl_Window *win)
Destroys the window.

- void **ewl_engine_window_dialog_set** (Ewl_Window *win)
Sets the window as a dialog.
- void **ewl_engine_window_geometry_get** (Ewl_Window *win, int root, int *width, int *height)
Retrieves the window geometry.
- void **ewl_engine_window_hide** (Ewl_Window *win)
Hides the window.
- void **ewl_engine_window_lower** (Ewl_Window *win)
Lowers the window.
- void **ewl_engine_window_min_max_size_set** (Ewl_Window *win)
Sets the window min/max size.
- void **ewl_engine_window_move** (Ewl_Window *win)
Moves the window.
- void **ewl_engine_window_name_class_set** (Ewl_Window *win)
Sets the name and class of the window.
- void **ewl_engine_window_new** (Ewl_Window *win)
Creates a new window.
- void **ewl_engine_window_raise** (Ewl_Window *win)
Raises the window.
- void **ewl_engine_window_resize** (Ewl_Window *win)
Resizes the window.
- void **ewl_engine_window_selection_text_set** (Ewl_Window *win, const char *txt)
Sets the window selection text.
- void **ewl_engine_window_show** (Ewl_Window *win)
Shows the window.
- void **ewl_engine_window_states_set** (Ewl_Window *win)
Sets the window state values.
- void **ewl_engine_window_title_set** (Ewl_Window *win)
Sets the title of the window.
- void **ewl_engine_window_transient_for** (Ewl_Window *win)
Sets the window transient.
- int **ewl_engines_init** (void)
Initialises the engine system.

- void **ewl_engines_shutdown** (void)

Shutdown the engine system.

8.70.1 Define Documentation

8.70.1.1 `#define EWL_ENGINE(engine) ((Ewl_Engine *)engine)`

8.70.2 Typedef Documentation

8.70.2.1 `typedef struct Ewl_Engine Ewl_Engine`

`Ewl_Engine` (p. 633)

8.70.2.2 `typedef enum Ewl_Engine_Canvas_Hooks Ewl_Engine_Canvas_Hooks`

8.70.2.3 `typedef void(*) Ewl_Engine_Cb_Canvas_Damage_Add(Ewl_Embed *embed, int x, int y, int w, int h)`

8.70.2.4 `typedef void(*) Ewl_Engine_Cb_Canvas_Freeze(Ewl_Embed *embed)`

8.70.2.5 `typedef void(*) Ewl_Engine_Cb_Canvas_Output_Set(Ewl_Embed *emb, int x, int y, int width, int height)`

8.70.2.6 `typedef void(*) Ewl_Engine_Cb_Canvas_Render(Ewl_Embed *embed)`

8.70.2.7 `typedef void(*) Ewl_Engine_Cb_Canvas_Setup(Ewl_Window *win, int debug)`

Setup the render canvas

8.70.2.8 `typedef void(*) Ewl_Engine_Cb_Canvas_Thaw(Ewl_Embed *embed)`

8.70.2.9 `typedef int(*) Ewl_Engine_Cb_Keyboard_Grab(Ewl_Window *win)`

Set the keyboard grab

8.70.2.10 `typedef void(*) Ewl_Engine_Cb_Keyboard_Ungrab(Ewl_Window *win)`

Set the keyboard ungrab

8.70.2.11 `typedef int(*) Ewl_Engine_Cb_Pointer_Data_New(Ewl_Embed
 *embed, unsigned int *data, int w, int h)`

8.70.2.12 `typedef void(*) Ewl_Engine_Cb_Pointer_Free(Ewl_Embed *embed,
 int pointer)`

8.70.2.13 `typedef int(*) Ewl_Engine_Cb_Pointer_Get(Ewl_Embed *embed)`

8.70.2.14 `typedef int(*) Ewl_Engine_Cb_Pointer_Grab(Ewl_Window *win)`

Set the pointer grab

8.70.2.15 `typedef void(*) Ewl_Engine_Cb_Pointer_Set(Ewl_Embed *embed, int
 pointer)`

8.70.2.16 `typedef void(*) Ewl_Engine_Cb_Pointer_Ungrab(Ewl_Window *win)`

Set the pointer ungrab

- 8.70.2.17 `typedef void(*) Ewl_Engine_Cb_Theme_Clip_Add(Ewl_Embed *embed)`
- 8.70.2.18 `typedef void(*) Ewl_Engine_Cb_Theme_Clip_Clipees_Get(void *clip)`
- 8.70.2.19 `typedef void(*) Ewl_Engine_Cb_Theme_Clip_Color_Set(void *clip, int r, int g, int b, int a)`
- 8.70.2.20 `typedef void(*) Ewl_Engine_Cb_Theme_Clip_Del(void *clip)`
- 8.70.2.21 `typedef char(*) Ewl_Engine_Cb_Theme_Data_Get(char *path, char *key)`
- 8.70.2.22 `typedef void(*) Ewl_Engine_Cb_Theme_Freeze()`
- 8.70.2.23 `typedef void(*) Ewl_Engine_Cb_Theme_Object_Add(Ewl_Embed *embed)`
- 8.70.2.24 `typedef void(*) Ewl_Engine_Cb_Theme_Object_Clip_Set(void *obj, void *clip)`
- 8.70.2.25 `typedef void(*) Ewl_Engine_Cb_Theme_Object_Clip_Unset(void *obj)`
- 8.70.2.26 `typedef void(*) Ewl_Engine_Cb_Theme_Object_Color_Class_Set(char *name, int r, int g, int b, int r2, int g2, int b2, int r3, int g3, int b3)`
- 8.70.2.27 `typedef void(*) Ewl_Engine_Cb_Theme_Object_Del(void *obj)`
- 8.70.2.28 `typedef void(*) Ewl_Engine_Cb_Theme_Object_File_Set(void *obj, char *path, char *group)`
- 8.70.2.29 `typedef void(*) Ewl_Engine_Cb_Theme_Object_Hide(void *obj)`
- 8.70.2.30 `typedef void(*) Ewl_Engine_Cb_Theme_Object_Layer_Update(Ewl_Widget *w)`
- 8.70.2.31 `typedef void(*) Ewl_Engine_Cb_Theme_Object_Load_Error(void *obj)`
- 8.70.2.32 `typedef void(*) Ewl_Engine_Cb_Theme_Object_Max_Size_Get(void *obj, int *w, int *h)`
- 8.70.2.33 `typedef void(*) Ewl_Engine_Cb_Theme_Object_Min_Size_Get(void *obj, int *w, int *h)`
- 8.70.2.34 `typedef void(*) Ewl_Engine_Cb_Theme_Object_Move(void *obj, int x, int y)`
- 8.70.2.35 `typedef void(*) Ewl_Engine_Cb_Theme_Object_Part_Text_Set(void *obj, char *part, char *text)`
- 8.70.2.36 `typedef void(*) Ewl_Engine_Cb_Theme_Object_Resize(void *obj, int w, int h)`
- 8.70.2.37 `typedef void(*) Ewl_Engine_Cb_Theme_Object_Show(void *obj)`
- 8.70.2.38 `typedef void(*) Ewl_Engine_Cb_Theme_Object_Signal_Send(void *obj, char *state, char *source)`

8.70.2.43 `typedef void(*) Ewl_Engine_Cb_Window_Desktop_Size_Get(Ewl_Embed *embed, int *w, int *h)`

Get the desktop size

8.70.2.44 `typedef void(*) Ewl_Engine_Cb_Window_Destroy(Ewl_Window *win)`

Destroy the window

8.70.2.45 `typedef void(*) Ewl_Engine_Cb_Window_Dialog_Set(Ewl_Window *win)`

Set the dialog setting of the window

8.70.2.46 `typedef void(*) Ewl_Engine_Cb_Window_Dnd_Aware_Set(Ewl_Embed *embed)`

Set the embed dnd aware

8.70.2.47 `typedef int(*) Ewl_Engine_Cb_Window_Dnd_Drag_Data_Send(Ewl_Embed *embed, void *handle, void *data, int size)`

Send the drag data

8.70.2.48 `typedef void(*) Ewl_Engine_Cb_Window_Dnd_Drag_Drop(Ewl_Embed *embed)`

End the drag process with a drop event

8.70.2.49 `typedef void(*) Ewl_Engine_Cb_Window_Dnd_Drag_Start(Ewl_Embed *embed)`

Start the drag process

8.70.2.50 `typedef void(*) Ewl_Engine_Cb_Window_Dnd_Drag_Types_Set(Ewl_Embed *embed, char **types, unsigned int num)`

Set the drag types

8.70.2.51 `typedef void(*) Ewl_Engine_Cb_Window_Geometry_Get(Ewl_Window *win, int *width, int *height)`

Get the window geometry

8.70.2.52 `typedef void(*) Ewl_Engine_Cb_Window_Hide(Ewl_Window *win)`

Hide the window

8.70.2.53 `typedef void(*) Ewl_Engine_Cb_Window_Lower(Ewl_Window *win)`

Lower the window

8.70.2.54 `typedef void(*) Ewl_Engine_Cb_Window_Min_Max_Size_Set(Ewl_Window *win)`

Set the min and max sizes

8.70.2.55 `typedef void(*) Ewl_Engine_Cb_Window_Move(Ewl_Window *win)`

Move the window to the given location

8.70.2.56 `typedef void(*) Ewl_Engine_Cb_Window_Name_Class_Set(Ewl_Window *win)`

Set the window name/class

8.70.2.57 `typedef void(*) Ewl_Engine_Cb_Window_New(Ewl_Window *win)`

Internal engine hook typedefs, you only need these if you're writing an engine. Create the window

8.70.2.58 `typedef void(*) Ewl_Engine_Cb_Window_Raise(Ewl_Window *win)`

Raise the window

8.70.2.59 `typedef void(*) Ewl_Engine_Cb_Window_Resize(Ewl_Window *win)`

Resize the window

8.70.2.60 `typedef void(*) Ewl_Engine_Cb_Window_Selection_Text_Set(Ewl_Window *win, const char *txt)`

Set the selection text

8.70.2.61 `typedef void(*) Ewl_Engine_Cb_Window_Show(Ewl_Window *win)`

Show the window

8.70.2.62 `typedef void(*) Ewl_Engine_Cb_Window_States_Set(Ewl_Window *win)`

Set the window modal

8.70.2.63 `typedef void(*) Ewl_Engine_Cb_Window_Title_Set(Ewl_Window *win)`

Set the window title

8.70.2.64 `typedef void(*) Ewl_Engine_Cb_Window_Transient_For(Ewl_Window *win)`

Set the window transient

8.70.2.65 `typedef struct Ewl_Engine_Info Ewl_Engine_Info`

`Ewl_Engine_Info` (p. 634)

8.70.2.66 `typedef enum Ewl_Engine_Pointer_Hooks Ewl_Engine_Pointer_Hooks`

8.70.2.67 `typedef enum Ewl_Engine_Theme_Hooks Ewl_Engine_Theme_Hooks`

8.70.2.68 `typedef enum Ewl_Engine_Window_Hooks Ewl_Engine_Window_Hooks`

8.70.3 Enumeration Type Documentation

8.70.3.1 `enum Ewl_Engine_Canvas_Hooks`

Enumerator:

EWL_ENGINE_CANVAS_SETUP Setup the render canvas
EWL_ENGINE_CANVAS_OUTPUT_SET Set the canvas size
EWL_ENGINE_CANVAS_RENDER
EWL_ENGINE_CANVAS_FREEZE
EWL_ENGINE_CANVAS_THAW
EWL_ENGINE_CANVAS_DAMAGE_ADD
EWL_ENGINE_CANVAS_MAX

8.70.3.2 `enum Ewl_Engine_Pointer_Hooks`

Enumerator:

EWL_ENGINE_POINTER_DATA_NEW Create pointer from ARGB data
EWL_ENGINE_POINTER_FREE Free a created pointer
EWL_ENGINE_POINTER_GET Get the current pointer
EWL_ENGINE_POINTER_SET Set the current pointer
EWL_ENGINE_POINTER_MAX

8.70.3.3 enum Ewl_Engine_Theme_Hooks

Enumerator:

```

EWL_ENGINE_THEME_FREEZE
EWL_ENGINE_THEME_THAW
EWL_ENGINE_THEME_DATA_GET
EWL_ENGINE_THEME_WIDGET_GROUP
EWL_ENGINE_THEME_OBJECT_ADD
EWL_ENGINE_THEME_OBJECT_DEL
EWL_ENGINE_THEME_OBJECT_MOVE
EWL_ENGINE_THEME_OBJECT_RESIZE
EWL_ENGINE_THEME_OBJECT_SHOW
EWL_ENGINE_THEME_OBJECT_HIDE
EWL_ENGINE_THEME_OBJECT_CLIP_SET
EWL_ENGINE_THEME_OBJECT_CLIP_UNSET
EWL_ENGINE_THEME_OBJECT_FILE_SET
EWL_ENGINE_THEME_OBJECT_LOAD_ERROR
EWL_ENGINE_THEME_OBJECT_MIN_SIZE_GET
EWL_ENGINE_THEME_OBJECT_MAX_SIZE_GET
EWL_ENGINE_THEME_OBJECT_SIGNAL_SEND
EWL_ENGINE_THEME_OBJECT_PART_TEXT_SET
EWL_ENGINE_THEME_OBJECT_COLOR_CLASS_SET
EWL_ENGINE_THEME_CLIP_ADD
EWL_ENGINE_THEME_CLIP_DEL
EWL_ENGINE_THEME_CLIP_CLIPEES_GET
EWL_ENGINE_THEME_CLIP_COLOR_SET
EWL_ENGINE_THEME_WIDGET_STACK_ADD
EWL_ENGINE_THEME_WIDGET_LAYER_UPDATE
EWL_ENGINE_THEME_MAX

```

8.70.3.4 enum Ewl_Engine_Window_Hooks

Enumerator:

```

EWL_ENGINE_WINDOW_NEW    Create the window
EWL_ENGINE_WINDOW_DESTROY Destroy the window
EWL_ENGINE_WINDOW_MOVE   Move the window to the given location
EWL_ENGINE_WINDOW_RESIZE Resize the window
EWL_ENGINE_WINDOW_MIN_MAX_SIZE_SET Set the min and max sizes

EWL_ENGINE_WINDOW_SHOW   Show the window
EWL_ENGINE_WINDOW_HIDE   Hide the window
EWL_ENGINE_WINDOW_TITLE_SET Set the window title

```

EWL_ENGINE_WINDOW_NAME_CLASS_SET Set the window name/class
EWL_ENGINE_WINDOW_BORDERLESS_SET Set the borderless state of the window
EWL_ENGINE_WINDOW_DIALOG_SET Set the dialog setting of the window
EWL_ENGINE_WINDOW_STATES_SET Set the window state flags
EWL_ENGINE_WINDOW_TRANSIENT_FOR Set the window transient
EWL_ENGINE_WINDOW_RAISE Raise the window
EWL_ENGINE_WINDOW_LOWER Lower the window
EWL_ENGINE_WINDOW_KEYBOARD_GRAB Set the keyboard grab
EWL_ENGINE_WINDOW_KEYBOARD_UNGRAB Set the keyboard ungrab
EWL_ENGINE_WINDOW_POINTER_GRAB Set the pointer grab
EWL_ENGINE_WINDOW_POINTER_UNGRAB Set the pointer ungrab
EWL_ENGINE_WINDOW_SELECTION_TEXT_SET Set the selection text
EWL_ENGINE_WINDOW_GEOMETRY_GET Get the window geometry
EWL_ENGINE_WINDOW_DND_AWARE_SET Set the window dnd aware
EWL_ENGINE_WINDOW_DESKTOP_SIZE_GET Get the size of the desktop

EWL_ENGINE_WINDOW_DND_DRAG_TYPES_SET Set window drag types
EWL_ENGINE_WINDOW_DND_DRAG_START Begin DND drag process
EWL_ENGINE_WINDOW_DND_DRAG_DROP Complete DND drag process
EWL_ENGINE_WINDOW_DND_DRAG_DATA_SEND Send data after request
EWL_ENGINE_WINDOW_MAX

8.70.4 Function Documentation

8.70.4.1 void ewl_engine_canvas_damage_add (Ewl_Embed * *embed*, int *x*, int *y*, int *w*, int *h*)

8.70.4.2 void ewl_engine_canvas_freeze (Ewl_Embed * *embed*)

Freezes the canvas.

Parameters:

embed,: the embed to work with

Returns:

Returns no value

8.70.4.3 void ewl_engine_canvas_output_set (Ewl_Embed * *emb*, int *x*, int *y*, int *width*, int *height*)

Sets up the canvas.

Parameters:

emb,: The embed to work with
x,: The x coord to set
y,: The y coord to set
width,: The width to set
height,: The height to set

Returns:

Returns no value

8.70.4.4 void ewl_engine_canvas_render (Ewl_Embed * *embed*)

Renders the canvas.

Parameters:

embed,: the embed to work with

Returns:

Returns no value

8.70.4.5 void ewl_engine_canvas_setup (Ewl_Window * *win*, int *debug*)

Sets up the canvas.

Parameters:

win,: the window to work with
debug,: The debug setting

Returns:

Returns no value

8.70.4.6 void ewl_engine_canvas_thaw (Ewl_Embed * *embed*)

Thaws the canvas.

Parameters:

embed,: the embed to work with

Returns:

Returns no value

8.70.4.7 void ewl_engine_desktop_size_get (Ewl_Embed * *embed*, int * *w*, int * *h*)

Sets the dnd awareness.

Parameters:

embed,: the embed to work with
w,: a pointer where the width of the desktop will be stored
h,: a pointer where the height pf the desktop will be stored

Returns:

Returns no value

8.70.4.8 void ewl_engine_embed_dnd_aware_set (Ewl_Embed * *embed*)

Sets the dnd awareness.

Parameters:

embed,: the embed to work with

Returns:

Returns no value

8.70.4.9 int ewl_engine_embed_dnd_drag_data_send (Ewl_Embed * *embed*, void * *handle*, void * *data*, int *size*)

Sends dnd data to the drop recipient.

Parameters:

embed,: the embed to work with
handle,: The DND handle to work with
data,: The data to send
size,: The size of the data being sent

Returns:

Returns no value

8.70.4.10 void ewl_engine_embed_dnd_drag_drop (Ewl_Embed * *embed*)

Completes the dnd process on an embed.

Parameters:

embed,: the embed to work with

Returns:

Returns no value

8.70.4.11 void ewl_engine_embed_dnd_drag_start (Ewl_Embed * *embed*)

Starts the dnd process on an embed.

Parameters:

embed,: the embed to work with

Returns:

Returns no value

**8.70.4.12 void ewl_engine_embed_dnd_drag_types_set (Ewl_Embed * *embed*,
char ** *types*, unsigned int *num*)**

Sets the dnd types provided by this embed.

Parameters:

embed,: the embed to work with

types,: The dnd types to set

num,: The number of types

Returns:

Returns no value

8.70.4.13 void ewl_engine_engine_shutdown (Ewl_Window * *win*)**8.70.4.14 void ewl_engine_event_handlers_init (void)****8.70.4.15 void ewl_engine_event_handlers_shutdown (void)****8.70.4.16 int ewl_engine_keyboard_grab (Ewl_Window * *win*)**

Sets the keyboard grab.

Parameters:

win,: the window to work with

Returns:

The keyboard grab value

8.70.4.17 void ewl_engine_keyboard_ungrab (Ewl_Window * *win*)

Sets the keyboard ungrab.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.18 `Ecore_List* ewl_engine_names_get (void)`

Retrieves a list of available engine names.

Returns:

Returns a list of engine names

8.70.4.19 `Ewl_Engine* ewl_engine_new (const char * name, int * argc, char **
argv)`

Retrieves, or creates the given **Ewl_Engine** (p. 633), or NULL on failure.

Parameters:

name,: The name of the engine to create

argc,: Arguments count

argv,: Arguments

Returns:

Returns the **Ewl_Engine** (p. 633) or NULL on failure

8.70.4.20 `int ewl_engine_pointer_data_new (Ewl_Engine * engine, unsigned int
* data, int w, int h)`

Creates a new pointer from ARGB data.

Returns:

Returns a pointer id on success, zero on failure.

8.70.4.21 `void ewl_engine_pointer_free (Ewl_Engine * engine, int pointer)`

Free the identified pointer.

Returns:

Returns no value.

8.70.4.22 `int ewl_engine_pointer_get (Ewl_Engine * engine)`

Retrieve the currently used pointer.

Returns:

Returns a pointer id on success, zero on failure.

8.70.4.23 int ewl_engine_pointer_grab (Ewl_Window * *win*)

Sets the pointer grab.

Parameters:

win,: the window to work with

Returns:

Returns the pointer grab value

8.70.4.24 void ewl_engine_pointer_set (Ewl_Embed * *embed*, int *pointer*)

Change the current pointer to the one identified.

Returns:

Returns no value.

8.70.4.25 void ewl_engine_pointer_ungrab (Ewl_Window * *win*)

Sets the pointer ungrab.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.26 void ewl_engine_window_borderless_set (Ewl_Window * *win*)

Sets the window borderless.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.27 void ewl_engine_window_destroy (Ewl_Window * *win*)

Destroys the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.28 void ewl_engine_window_dialog_set (Ewl_Window * *win*)

Sets the window as a dialog.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.29 void ewl_engine_window_geometry_get (Ewl_Window * *win*, int *root*, int * *width*, int * *height*)

Retrieves the window geometry.

Parameters:

win,: the window to work with

root,: The root window to get the height for

width,: Where to store the width

height,: Where to store the height

Returns:

Returns no value

We dont' check win == null in case their querying after the root window

8.70.4.30 void ewl_engine_window_hide (Ewl_Window * *win*)

Hides the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.31 void ewl_engine_window_lower (Ewl_Window * *win*)

Lowers the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.32 void ewl_engine_window_min_max_size_set (Ewl_Window * *win*)

Sets the window min/max size.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.33 void ewl_engine_window_move (Ewl_Window * *win*)

Moves the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.34 void ewl_engine_window_name_class_set (Ewl_Window * *win*)

Sets the name and class of the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.35 void ewl_engine_window_new (Ewl_Window * *win*)

Creates a new window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.36 void ewl_engine_window_raise (Ewl_Window * *win*)

Raises the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.37 void ewl_engine_window_resize (Ewl_Window * *win*)

Resizes the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

**8.70.4.38 void ewl_engine_window_selection_text_set (Ewl_Window * *win*,
const char * *txt*)**

Sets the window selection text.

Parameters:

win,: the window to work with

txt,: The text to set as the selection

Returns:

Returns no value

8.70.4.39 void ewl_engine_window_show (Ewl_Window * *win*)

Shows the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.40 void ewl_engine_window_states_set (Ewl_Window * *win*)

Sets the window state values.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.41 void ewl_engine_window_title_set (Ewl_Window * *win*)

Sets the title of the window.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.42 void ewl_engine_window_transient_for (Ewl_Window * *win*)

Sets the window transient.

Parameters:

win,: the window to work with

Returns:

Returns no value

8.70.4.43 int ewl_engines_init (void)

Initialises the engine system.

Returns:

Returns no value

8.70.4.44 void ewl_engines_shutdown (void)

Shutdown the engine system.

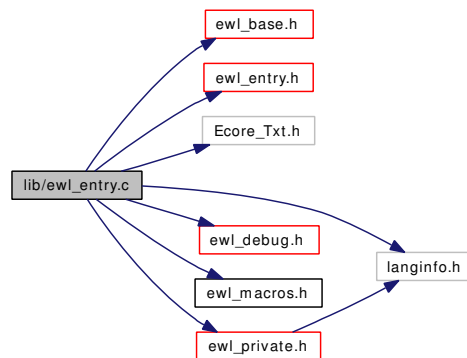
Returns:

Returns no value

8.71 lib/ewl_entry.c File Reference

```
#include "ewl_base.h"
#include "ewl_entry.h"
#include <Ecore_Txt.h>
#include <langinfo.h>
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_entry.c:



Functions

- void **ewl_entry_cb_configure** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_entry_cb_disable** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_entry_cb_dnd_data** (Ewl_Widget *w, void *ev, void *data __UNUSED__)
- void **ewl_entry_cb_dnd_position** (Ewl_Widget *w, void *ev, void *data __UNUSED__)
- void **ewl_entry_cb_enable** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_entry_cb_focus_in** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_entry_cb_focus_out** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_entry_cb_key_down** (Ewl_Widget *w, void *ev, void *data __UNUSED__)
- void **ewl_entry_cb_mouse_down** (Ewl_Widget *w, void *ev, void *data __UNUSED__)
- void **ewl_entry_cb_mouse_move** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_entry_cb_mouse_up** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)

- `int ewl_entry_cursor_init (Ewl_Entry_Cursor *c, Ewl_Entry *parent)`
*Initializes an **Ewl_Entry_Cursor** (p. 637) to default values.*
- `void ewl_entry_cursor_move_down (Ewl_Entry *e)`
Moves the cursor down one line from its current position.
- `void ewl_entry_cursor_move_left (Ewl_Entry *e)`
Moves the cursor to the left of it's current position.
- `void ewl_entry_cursor_move_right (Ewl_Entry *e)`
Moves the cursor to the right of its current position.
- `void ewl_entry_cursor_move_up (Ewl_Entry *e)`
Moves the cursor up one line from its current position.
- `Ewl_Widget * ewl_entry_cursor_new (Ewl_Entry *parent)`
*Creates and initializes a new **Ewl_Entry_Cursor** (p. 637) widget.*
- `unsigned int ewl_entry_cursor_position_get (Ewl_Entry_Cursor *c)`
*Retrieves the current position of the **Ewl_Entry_Cursor** (p. 637).*
- `void ewl_entry_cursor_position_set (Ewl_Entry_Cursor *c, unsigned int pos)`
Set the current cursor position.
- `void ewl_entry_delete_left (Ewl_Entry *e)`
Deletes the character to the left of the cursor.
- `void ewl_entry_delete_right (Ewl_Entry *e)`
Deletes the character to the right of the cursor.
- `unsigned int ewl_entry_editable_get (Ewl_Entry *e)`
Retrieve if the entry is editable or not.
- `void ewl_entry_editable_set (Ewl_Entry *e, unsigned int editable)`
Set if the entry is editable or not.
- `int ewl_entry_init (Ewl_Entry *e)`
*Initializes an **Ewl_Entry** (p. 635) widget to default values.*
- `unsigned int ewl_entry_multiline_get (Ewl_Entry *e)`
Retrieve if the entry is multiline or not.
- `void ewl_entry_multiline_set (Ewl_Entry *e, unsigned int multiline)`
Set if the entry is multiline or not.
- `Ewl_Widget * ewl_entry_new (void)`
*Create and return a new **Ewl_Entry** (p. 635) widget.*
- `unsigned int ewl_entry_selection_clear (Ewl_Entry *e)`
Clear the current selection in the entry.

8.71.1 Function Documentation

- 8.71.1.1 `void ewl_entry_cb_configure (Ewl_Widget * w, void *ev
__ UNUSED __, void *data __ UNUSED __)`
- 8.71.1.2 `void ewl_entry_cb_disable (Ewl_Widget * w, void *ev __ UNUSED __,
void *data __ UNUSED __)`
- 8.71.1.3 `void ewl_entry_cb_dnd_data (Ewl_Widget * w, void * ev, void *data
__ UNUSED __)`
- 8.71.1.4 `void ewl_entry_cb_dnd_position (Ewl_Widget * w, void * ev, void
*data __ UNUSED __)`
- 8.71.1.5 `void ewl_entry_cb_enable (Ewl_Widget * w, void *ev __ UNUSED __,
void *data __ UNUSED __)`
- 8.71.1.6 `void ewl_entry_cb_focus_in (Ewl_Widget * w, void *ev
__ UNUSED __, void *data __ UNUSED __)`
- 8.71.1.7 `void ewl_entry_cb_focus_out (Ewl_Widget * w, void *ev
__ UNUSED __, void *data __ UNUSED __)`
- 8.71.1.8 `void ewl_entry_cb_key_down (Ewl_Widget * w, void * ev, void *data
__ UNUSED __)`
- 8.71.1.9 `void ewl_entry_cb_mouse_down (Ewl_Widget * w, void * ev, void
*data __ UNUSED __)`
- 8.71.1.10 `void ewl_entry_cb_mouse_move (Ewl_Widget * w, void *ev
__ UNUSED __, void *data __ UNUSED __)`
- 8.71.1.11 `void ewl_entry_cb_mouse_up (Ewl_Widget * w, void *ev
__ UNUSED __, void *data __ UNUSED __)`

8.72 bin/tests/entry/ewl_entry.c File Reference

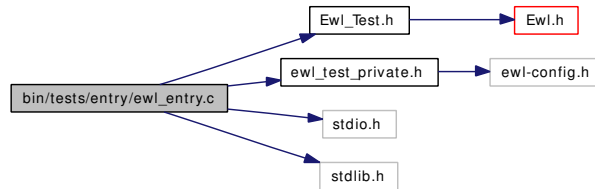
```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

Include dependency graph for ewl_entry.c:



Functions

- `void test_info (Ewl_Test *test)`

8.72.1 Function Documentation

8.72.1.1 `void test_info (Ewl_Test * test)`

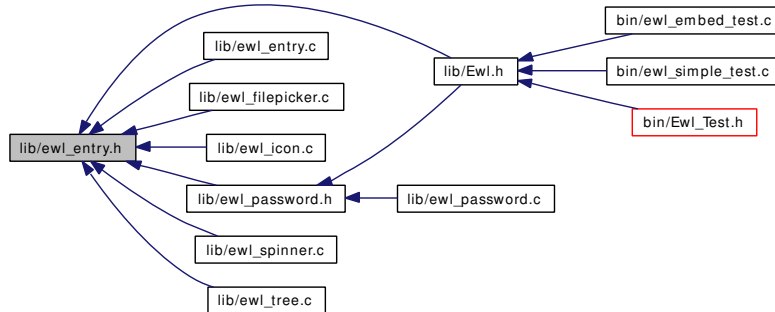
8.73 lib/ewl_entry.h File Reference

```
#include "ewl_text.h"
```

Include dependency graph for ewl_entry.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Entry**

*Inherits from **Ewl_Text** (p. 757) and extends to allow editing of the text.*

- struct **Ewl_Entry_Cursor**

*Inherit from **Ewl_Widget** (p. 785) and extend for the Entry cursor.*

Defines

- #define **EWL_ENTRY(entry)** ((**Ewl_Entry** *) entry)
- #define **EWL_ENTRY_CURSOR(cursor)** ((**Ewl_Entry_Cursor** *) cursor)
- #define **EWL_ENTRY_CURSOR_IS(w)** (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_ENTRY_CURSOR_TYPE**))
- #define **EWL_ENTRY_CURSOR_TYPE** "cursor"
- #define **EWL_ENTRY_IS(w)** (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_ENTRY_TYPE**))
- #define **EWL_ENTRY_TYPE** "entry"

Typedefs

- typedef **Ewl_Entry** **Ewl_Entry**
- typedef **Ewl_Entry_Cursor** **Ewl_Entry_Cursor**

Functions

- void **ewl_entry_cb_configure** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_entry_cb_disable** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_entry_cb_dnd_data** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_entry_cb_dnd_position** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_entry_cb_enable** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_entry_cb_focus_in** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_entry_cb_focus_out** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_entry_cb_key_down** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_entry_cb_mouse_down** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_entry_cb_mouse_move** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_entry_cb_mouse_up** (Ewl_Widget *w, void *ev, void *data)
- int **ewl_entry_cursor_init** (Ewl_Entry_Cursor *c, Ewl_Entry *parent)
*Initializes an **Ewl_Entry_Cursor** (p. 637) to default values.*
- void **ewl_entry_cursor_move_down** (Ewl_Entry *e)
Moves the cursor down one line from its current position.
- void **ewl_entry_cursor_move_left** (Ewl_Entry *e)
Moves the cursor to the left of it's current position.
- void **ewl_entry_cursor_move_right** (Ewl_Entry *e)
Moves the cursor to the right of its current position.
- void **ewl_entry_cursor_move_up** (Ewl_Entry *e)
Moves the cursor up one line from its current position.
- Ewl_Widget * **ewl_entry_cursor_new** (Ewl_Entry *parent)
*Creates and initializes a new **Ewl_Entry_Cursor** (p. 637) widget.*
- unsigned int **ewl_entry_cursor_position_get** (Ewl_Entry_Cursor *c)
*Retrieves the current position of the **Ewl_Entry_Cursor** (p. 637).*
- void **ewl_entry_cursor_position_set** (Ewl_Entry_Cursor *c, unsigned int pos)
Set the current cursor position.
- void **ewl_entry_delete_left** (Ewl_Entry *e)
Deletes the character to the left of the cursor.
- void **ewl_entry_delete_right** (Ewl_Entry *e)
Deletes the character to the right of the cursor.
- unsigned int **ewl_entry_editable_get** (Ewl_Entry *e)
Retrieve if the entry is editable or not.
- void **ewl_entry_editable_set** (Ewl_Entry *e, unsigned int editable)
Set if the entry is editable or not.
- int **ewl_entry_init** (Ewl_Entry *e)

*Initializes an **Ewl_Entry** (p. 635) widget to default values.*

- unsigned int **ewl_entry_multiline_get** (**Ewl_Entry** *e)
Retrieve if the entry is multiline or not.
- void **ewl_entry_multiline_set** (**Ewl_Entry** *e, unsigned int multiline)
Set if the entry is multiline or not.
- **Ewl_Widget** * **ewl_entry_new** (void)
*Create and return a new **Ewl_Entry** (p. 635) widget.*
- unsigned int **ewl_entry_selection_clear** (**Ewl_Entry** *e)
Clear the current selection in the entry.

8.74 lib/ewl_enums.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define EWL_FLAG_FILL_NORMAL (EWL_FLAG_FILL_FILL)`
- `#define EWL_FLAGS_ALIGN_MASK`
- `#define EWL_FLAGS_FILL_MASK`
- `#define EWL_FLAGS_PROPERTY_MASK`
- `#define EWL_FLAGS_QUEUED_MASK`
- `#define EWL_FLAGS_STATE_MASK`
- `#define EWL_FLAGS_VISIBLE_MASK`
- `#define EWL_POSITION_MASK (0xf)`

Typedefs

- `typedef enum Ewl_Attach_Data_Type Ewl_Attach_Data_Type`
- `typedef enum Ewl_Attach_Type Ewl_Attach_Type`
- `typedef enum Ewl_Callback_Type Ewl_Callback_Type`
- `typedef enum Ewl_Color_Mode Ewl_Color_Mode`
- `typedef enum Ewl_Event_Notify Ewl_Event_Notify`
- `typedef enum Ewl_Filelist_Event_Type Ewl_Filelist_Event_Type`
- `typedef enum Ewl_Freebox_Layout_Type Ewl_Freebox_Layout_Type`
- `typedef enum Ewl_Grid_Resize_Type Ewl_Grid_Resize_Type`
- `typedef enum Ewl_Histogram_Channel Ewl_Histogram_Channel`
- `typedef enum Ewl_Icon_Part Ewl_Icon_Part`
- `typedef enum Ewl_Icon_Type Ewl_Icon_Type`
- `typedef enum Ewl_Image_Type Ewl_Image_Type`
- `typedef enum Ewl_Key_Modifiers Ewl_Key_Modifiers`
- `typedef enum Ewl_Media_Module_Type Ewl_Media_Module_Type`
- `typedef enum Ewl_Mouse_Cursor_Type Ewl_Mouse_Cursor_Type`
- `typedef enum Ewl_Notebook_Flags Ewl_Notebook_Flags`
- `typedef enum Ewl_Orientation Ewl_Orientation`
- `typedef enum Ewl_Popup_Type Ewl_Popup_Type`
- `typedef enum Ewl_Position Ewl_Position`
- `typedef enum Ewl_Rotate Ewl_Rotate`
- `typedef enum Ewl_Scrollpane_Flags Ewl_Scrollpane_Flags`
- `typedef enum Ewl_Selection_Mode Ewl_Selection_Mode`
- `typedef enum Ewl_Selection_Type Ewl_Selection_Type`
- `typedef enum Ewl_Sort_Direction Ewl_Sort_Direction`
- `typedef enum Ewl_Spectrum_Type Ewl_Spectrum_Type`
- `typedef enum Ewl_State_Type Ewl_State_Type`
- `typedef enum Ewl_Stock_Type Ewl_Stock_Type`
- `typedef enum Ewl_Text_Context_Mask Ewl_Text_Context_Mask`
- `typedef enum Ewl_Text_Style Ewl_Text_Style`

- typedef enum Ewl_Text_Trigger_Type Ewl_Text_Trigger_Type
- typedef enum Ewl_Text_Wrap Ewl_Text_Wrap
- typedef enum Ewl_Tree_Node_Flags Ewl_Tree_Node_Flags
- typedef enum Ewl_Tree_Selection_Type Ewl_Tree_Selection_Type
- typedef enum Ewl_Window_Flags Ewl_Window_Flags

Enumerations

- enum Ewl_Attach_Data_Type { EWL_ATTACH_DATA_TYPE_TEXT, EWL_ATTACH_DATA_TYPE_WIDGET, EWL_ATTACH_DATA_TYPE_OTHER }
- enum Ewl_Attach_Type {
EWL_ATTACH_TYPE_TOOLTIP, EWL_ATTACH_TYPE_COLOR,
EWL_ATTACH_TYPE_NAME, EWL_ATTACH_TYPE_MOUSE_CURSOR,
EWL_ATTACH_TYPE_MOUSE_ARGB_CURSOR, EWL_ATTACH_TYPE_WIDGET_ASSOCIATION, EWL_ATTACH_TYPE_DND_DATA
}
- enum Ewl_Callback_Type {
EWL_CALLBACK_REALIZE, EWL_CALLBACK_UNREALIZE, EWL_CALLBACK_SHOW, EWL_CALLBACK_HIDE,
EWL_CALLBACK_OBSCURE, EWL_CALLBACK_REVEAL, EWL_CALLBACK_DESTROY, EWL_CALLBACK_CONFIGURE,
EWL_CALLBACK_REPARENT, EWL_CALLBACK_KEY_DOWN, EWL_CALLBACK_KEY_UP, EWL_CALLBACK_MOUSE_DOWN,
EWL_CALLBACK_MOUSE_UP, EWL_CALLBACK_MOUSE_MOVE, EWL_CALLBACK_MOUSE_WHEEL, EWL_CALLBACK_MOUSE_IN,
EWL_CALLBACK_MOUSE_OUT, EWL_CALLBACK_CLICKED, EWL_CALLBACK_FOCUS_IN, EWL_CALLBACK_FOCUS_OUT,
EWL_CALLBACK_VALUE_CHANGED, EWL_CALLBACK_STATE_CHANGED, EWL_CALLBACK_WIDGET_ENABLE, EWL_CALLBACK_WIDGET_DISABLE,
EWL_CALLBACK_MAX }
}
- enum Ewl_Color_Mode {
EWL_COLOR_MODE_RGB_RED, EWL_COLOR_MODE_RGB_GREEN, EWL_COLOR_MODE_RGB_BLUE, EWL_COLOR_MODE_HSV_HUE,
EWL_COLOR_MODE_HSV_SATURATION, EWL_COLOR_MODE_HSV_VALUE }
}
- enum Ewl_Event_Notify { EWL_CALLBACK_NOTIFY_NONE = 0, EWL_CALLBACK_NOTIFY_NOTIFY = 1, EWL_CALLBACK_NOTIFY_INTERCEPT = 2, EWL_CALLBACK_TYPE_DIRECT = 4 }
- enum Ewl_Filelist_Event_Type { EWL_FILELIST_EVENT_DIR_CHANGE, EWL_FILELIST_EVENT_SELECTION_CHANGE, EWL_FILELIST_EVENT_FILE_SELECTED }
- enum Ewl_Flags {
EWL_FLAG_ALIGN_CENTER = 0, EWL_FLAG_ALIGN_LEFT = 0x1,
EWL_FLAG_ALIGN_RIGHT = 0x2, EWL_FLAG_ALIGN_TOP = 0x4,
}

```

EWL_FLAG_ALIGN_BOTTOM = 0x8, EWL_FLAG_FILL_NONE = 0,
EWL_FLAG_FILL_HSHRINK = 0x10, EWL_FLAG_FILL_VSHRINK =
0x20,
EWL_FLAG_FILL_SHRINK, EWL_FLAG_FILL_HFILL = 0x40, EWL_
FLAG_FILL_VFILL = 0x80, EWL_FLAG_FILL_FILL = EWL_FLAG_FILL_
HFILL | EWL_FLAG_FILL_VFILL,
EWL_FLAG_FILL_ALL = EWL_FLAG_FILL_FILL | EWL_FLAG_FILL_
SHRINK, EWL_FLAG_VISIBLE_HIDDEN = 0, EWL_FLAG_VISIBLE_
SHOWN = 0x100, EWL_FLAG_VISIBLE_REALIZED = 0x200,
EWL_FLAG_VISIBLE_OBSCURED = 0x400, EWL_FLAG_VISIBLE_
NOCLIP = 0x800, EWL_FLAG_PROPERTY_RECURSIVE = 0x1000, EWL_
FLAG_PROPERTY_TOPLEVEL = 0x2000,
EWL_FLAG_PROPERTY_INTERNAL = 0x4000, EWL_FLAG_
PROPERTY_BLOCK_TAB_FOCUS = 0x8000, EWL_FLAG_PROPERTY_
FOCUSABLE = 0x10000, EWL_FLAG_PROPERTY_IN_TAB_LIST =
0x20000,
EWL_FLAG_PROPERTY_DND_TARGET = 0x40000, EWL_FLAG_
PROPERTY_DND_SOURCE = 0x80000, EWL_FLAG_QUEUED_
CSCHEDULED = 0x100000, EWL_FLAG_QUEUED_RSCHEDULED =
0x200000,
EWL_FLAG_QUEUED_DSCHEDULED = 0x400000, EWL_FLAG_
QUEUED_CPROCESS = 0x800000, EWL_FLAG_QUEUED_RPROCESS
= 0x1000000, EWL_FLAG_QUEUED_DPROCESS = 0x2000000,
EWL_FLAG_STATE_NORMAL = 0, EWL_FLAG_STATE_MOUSE_IN
= 0x4000000, EWL_FLAG_STATE_PRESSED = 0x8000000, EWL_
FLAG_STATE_FOCUSED = 0x10000000,
EWL_FLAG_STATE_DISABLED = 0x20000000, EWL_FLAG_STATE_DND
= 0x40000000, EWL_FLAG_STATE_DND_WAIT = 0x80000000 }
• enum Ewl_Freebox_Layout_Type { EWL_FREEBOX_LAYOUT_MANUAL,
EWL_FREEBOX_LAYOUT_COMPARATOR, EWL_FREEBOX_
LAYOUT_AUTO }
• enum Ewl_Grid_Resize_Type { EWL_GRID_RESIZE_NORMAL, EWL_
GRID_RESIZE_FIXED, EWL_GRID_RESIZE_RELATIVE, EWL_
GRID_RESIZE_NONE }
• enum Ewl_Histogram_Channel {
EWL_HISTOGRAM_CHANNEL_Y, EWL_HISTOGRAM_CHANNEL_R,
EWL_HISTOGRAM_CHANNEL_G, EWL_HISTOGRAM_CHANNEL_B,
EWL_HISTOGRAM_CHANNEL_MAX }
• enum Ewl_Icon_Part { EWL_ICON_PART_NONE, EWL_ICON_PART_
IMAGE, EWL_ICON_PART_LABEL }
• enum Ewl_Icon_Type { EWL_ICON_TYPE_SHORT, EWL_ICON_TYPE_
LONG }
• enum Ewl_Image_Type { EWL_IMAGE_TYPE_NORMAL, EWL_
IMAGE_TYPE_EDJE }
• enum Ewl_Key_Modifiers {
EWL_KEY_MODIFIER_SHIFT = 0x1, EWL_KEY_MODIFIER_CTRL =
0x2, EWL_KEY_MODIFIER_ALT = 0x4, EWL_KEY_MODIFIER_MOD =
0x8,
EWL_KEY_MODIFIER_WIN = 0x10 }

```

- enum Ewl_Media_Module_Type { EWL_MEDIA_MODULE_XINE, EWL_MEDIA_MODULE_GSTREAMER }
- enum Ewl_Mouse_Cursor_Type {
 - EWL_MOUSE_CURSOR_X = 0, EWL_MOUSE_CURSOR_ARROW = 2, EWL_MOUSE_CURSOR_BASED_ARROW_DOWN = 4, EWL_MOUSE_CURSOR_UP = 6,
 - EWL_MOUSE_CURSOR_BOAT = 8, EWL_MOUSE_CURSOR_BOTTOM_LEFT_CORNER = 12, EWL_MOUSE_CURSOR_BOTTOM_RIGHT_CORNER = 14, EWL_MOUSE_CURSOR_BOTTOM_SIDE = 16,
 - EWL_MOUSE_CURSOR_BOTTOM_TEE = 18, EWL_MOUSE_CURSOR_BOX_SPIRAL = 20, EWL_MOUSE_CURSOR_CENTER_PTR = 22, EWL_MOUSE_CURSOR_CIRCLE = 24,
 - EWL_MOUSE_CURSOR_CLOCK = 26, EWL_MOUSE_CURSOR_COFFEE_MUG = 28, EWL_MOUSE_CURSOR_CROSS = 30, EWL_MOUSE_CURSOR_CROSS_REVERSE = 32,
 - EWL_MOUSE_CURSOR_CROSSHAIR = 34, EWL_MOUSE_CURSOR_DIAMOND_CROSS = 36, EWL_MOUSE_CURSOR_DOT = 38, EWL_MOUSE_CURSOR_DOT_BOX_MASK = 40,
 - EWL_MOUSE_CURSOR_DOUBLE_ARROW = 42, EWL_MOUSE_CURSOR_DRAFT_LARGE = 44, EWL_MOUSE_CURSOR_DRAFT_SMALL = 46, EWL_MOUSE_CURSOR_DRAPED_BOX = 48,
 - EWL_MOUSE_CURSOR_EXCHANGE = 50, EWL_MOUSE_CURSOR_FLEUR = 52, EWL_MOUSE_CURSOR_GOBLER = 54, EWL_MOUSE_CURSOR_GUMBY = 56,
 - EWL_MOUSE_CURSOR_HAND1 = 58, EWL_MOUSE_CURSOR_HAND2 = 60, EWL_MOUSE_CURSOR_HEART = 62, EWL_MOUSE_CURSOR_ICON = 64,
 - EWL_MOUSE_CURSOR_IRON_CROSS = 66, EWL_MOUSE_CURSOR_LEFT_PTR = 68, EWL_MOUSE_CURSOR_LEFT_SIDE = 70, EWL_MOUSE_CURSOR_LEFT_TEE = 72,
 - EWL_MOUSE_CURSOR_LEFTBUTTON = 74, EWL_MOUSE_CURSOR_LL_ANGLE = 76, EWL_MOUSE_CURSOR_LR_ANGLE = 78, EWL_MOUSE_CURSOR_MAN = 80,
 - EWL_MOUSE_CURSOR_MIDDLEBUTTON = 82, EWL_MOUSE_CURSOR_MOUSE = 84, EWL_MOUSE_CURSOR_PENCIL = 86, EWL_MOUSE_CURSOR_PIRATE = 88,
 - EWL_MOUSE_CURSOR_PLUS = 90, EWL_MOUSE_CURSOR_QUESTION_ARROW = 92, EWL_MOUSE_CURSOR_RIGHT_PTR = 94, EWL_MOUSE_CURSOR_RIGHT_SIDE = 96,
 - EWL_MOUSE_CURSOR_RIGHT_TEE = 98, EWL_MOUSE_CURSOR_RIGHTBUTTON = 100, EWL_MOUSE_CURSOR_RTL_LOGO = 102, EWL_MOUSE_CURSOR_SAILBOAT = 104,
 - EWL_MOUSE_CURSOR_SB_DOWN_ARROW = 106, EWL_MOUSE_CURSOR_SB_H_DOUBLE_ARROW = 108, EWL_MOUSE_CURSOR_SB_LEFT_ARROW = 110, EWL_MOUSE_CURSOR_SB_RIGHT_ARROW = 112,
 - EWL_MOUSE_CURSOR_SB_UP_ARROW = 114, EWL_MOUSE_CURSOR_SB_V_DOUBLE_ARROW = 116, EWL_MOUSE_CURSOR_SHUTTLE = 118, EWL_MOUSE_CURSOR_SIZING = 120,

```

EWL_MOUSE_CURSOR_SPIDER = 122, EWL_MOUSE_CURSOR_
SPRAYCAN = 124, EWL_MOUSE_CURSOR_STAR = 126, EWL_MOUSE_
CURSOR_TARGET = 128,

EWL_MOUSE_CURSOR_TCROSS = 130, EWL_MOUSE_CURSOR_
TOP_LEFT_ARROW = 132, EWL_MOUSE_CURSOR_TOP_LEFT_
CORNER = 134, EWL_MOUSE_CURSOR_TOP_RIGHT_CORNER =
136,

EWL_MOUSE_CURSOR_TOP_SIDE = 138, EWL_MOUSE_CURSOR_
TOP_TEE = 140, EWL_MOUSE_CURSOR_TREK = 142, EWL_MOUSE_
CURSOR_UL_ANGLE = 144,

EWL_MOUSE_CURSOR_UMBRELLA = 146, EWL_MOUSE_CURSOR_
UR_ANGLE = 148, EWL_MOUSE_CURSOR_WATCH = 150, EWL_
MOUSE_CURSOR_XTERM = 152,

EWL_MOUSE_CURSOR_MAX = 153 }

• enum Ewl_Notebook_Flags { EWL_NOTEBOOK_FLAG_TABS_HIDDEN =
0x10 }

• enum Ewl_Orientation { EWL_ORIENTATION_HORIZONTAL, EWL_
ORIENTATION_VERTICAL }

• enum Ewl_Popup_Type { EWL_POPUP_TYPE_NONE, EWL_POPUP_
TYPE_MOUSE, EWL_POPUP_TYPE_MENU_HORIZONTAL, EWL_
POPUP_TYPE_MENU_VERTICAL }

• enum Ewl_Position { EWL_POSITION_LEFT = 0x1, EWL_POSITION_
RIGHT = 0x2, EWL_POSITION_TOP = 0x4, EWL_POSITION_BOTTOM
= 0x8 }

• enum Ewl_Rotate {

EWL_ROTATE_CW_90, EWL_ROTATE_180, EWL_ROTATE_CW_270,
EWL_ROTATE_CC_90,

EWL_ROTATE_CC_270 }

• enum Ewl_Scrollpane_Flags { EWL_SCROLLPANE_FLAG_NONE, EWL_
SCROLLPANE_FLAG_AUTO_VISIBLE, EWL_SCROLLPANE_FLAG_
ALWAYS_HIDDEN }

• enum Ewl_Selection_Mode { EWL_SELECTION_MODE_NONE, EWL_
SELECTION_MODE_SINGLE, EWL_SELECTION_MODE_MULTI }

• enum Ewl_Selection_Type { EWL_SELECTION_TYPE_INDEX, EWL_
SELECTION_TYPE_RANGE }

• enum Ewl_Sort_Direction { EWL_SORT_DIRECTION_NONE = 0,
EWL_SORT_DIRECTION_ASCENDING, EWL_SORT_DIRECTION_
DESCENDING, EWL_SORT_DIRECTION_MAX }

• enum Ewl_Spectrum_Type { EWL_SPECTRUM_TYPE_SQUARE, EWL_
SPECTRUM_TYPE_VERTICAL }

• enum Ewl_State_Type { EWL_STATE_TRANSIENT, EWL_STATE_
PERSISTENT }

• enum Ewl_Stock_Type {

EWL_STOCK_APPLY = 0, EWL_STOCK_ARROW_DOWN, EWL_
STOCK_ARROW_LEFT, EWL_STOCK_ARROW_RIGHT,

EWL_STOCK_ARROW_UP, EWL_STOCK_CANCEL, EWL_STOCK_
FASTFORWARD, EWL_STOCK_HOME,

EWL_STOCK_OK, EWL_STOCK_OPEN, EWL_STOCK_PAUSE, EWL_
STOCK_PLAY,

```

```

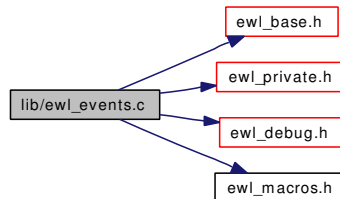
EWL_STOCK_QUIT, EWL_STOCK_REWIND, EWL_STOCK_SAVE,
EWL_STOCK_STOP,
EWL_STOCK_NONE }
• enum Ewl_Text_Context_Mask {
    EWL_TEXT_CONTEXT_MASK_NONE = 0x00, EWL_TEXT_CONTEXT_MASK_FONT = 0x01, EWL_TEXT_CONTEXT_MASK_SIZE = 0x02, EWL_TEXT_CONTEXT_MASK_STYLES = 0x04,
    EWL_TEXT_CONTEXT_MASK_ALIGN = 0x08, EWL_TEXT_CONTEXT_MASK_WRAP = 0x10, EWL_TEXT_CONTEXT_MASK_COLOR = 0x20, EWL_TEXT_CONTEXT_MASK_BG_COLOR = 0x40,
    EWL_TEXT_CONTEXT_MASK_GLOW_COLOR = 0x80, EWL_TEXT_CONTEXT_MASK_OUTLINE_COLOR = 0x100, EWL_TEXT_CONTEXT_MASK_SHADOW_COLOR = 0x200, EWL_TEXT_CONTEXT_MASK_STRIKETHROUGH_COLOR = 0x400,
    EWL_TEXT_CONTEXT_MASK_UNDERLINE_COLOR = 0x800, EWL_TEXT_CONTEXT_MASK_DOUBLE_UNDERLINE_COLOR = 0x1000 }
• enum Ewl_Text_Style {
    EWL_TEXT_STYLE_NONE = 0x00, EWL_TEXT_STYLE_UNDERLINE = 0x01, EWL_TEXT_STYLE_DOUBLE_UNDERLINE = 0x02, EWL_TEXT_STYLE_STRIKETHROUGH = 0x04,
    EWL_TEXT_STYLE_SHADOW = 0x08, EWL_TEXT_STYLE_SOFT_SHADOW = 0x10, EWL_TEXT_STYLE_FAR_SHADOW = 0x20, EWL_TEXT_STYLE_OUTLINE = 0x40,
    EWL_TEXT_STYLE_GLOW = 0x80 }
• enum Ewl_Text_Trigger_Type { EWL_TEXT_TRIGGER_TYPE_NONE, EWL_TEXT_TRIGGER_TYPE_SELECTION, EWL_TEXT_TRIGGER_TYPE_TRIGGER }
• enum Ewl_Text_Wrap { EWL_TEXT_WRAP_NONE = 0, EWL_TEXT_WRAP_CHAR, EWL_TEXT_WRAP_WORD }
• enum Ewl_Tree_Node_Flags { EWL_TREE_NODE_NOEXPAND = 0, EWL_TREE_NODE_COLLAPSED = 1, EWL_TREE_NODE_EXPANDED = 2 }
• enum Ewl_Tree_Selection_Type { EWL_TREE_SELECTION_TYPE_ROW, EWL_TREE_SELECTION_TYPE_CELL }
• enum Ewl_Window_Flags {
    EWL_WINDOW_BORDERLESS = 0x1, EWL_WINDOW_USER_CONFIGURE = 0x2, EWL_WINDOW_GRAB_POINTER = 0x4, EWL_WINDOW_GRAB_KEYBOARD = 0x8,
    EWL_WINDOW_OVERRIDE = 0x10, EWL_WINDOW_FULLSCREEN = 0x20, EWL_WINDOW_DIALOG = 0x40, EWL_WINDOW_MODAL = 0x80,
    EWL_WINDOW_TRANSIENT = 0x100, EWL_WINDOW_TRANSIENT_FOREIGN = 0x200, EWL_WINDOW_SKIP_TASKBAR = 0x400, EWL_WINDOW_SKIP_PAGER = 0x800,
    EWL_WINDOW_DEMANDS_ATTENTION = 0x1000 }

```


8.75 lib/ewl_events.c File Reference

```
#include "ewl_base.h"
#include "ewl_private.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
```

Include dependency graph for ewl_events.c:



Functions

- unsigned int **ewl_ev_modifiers_get** (void)
Retrieve the current mask of modifiers that are set.
- void **ewl_ev_modifiers_set** (unsigned int modifiers)
Sets the key modifiers to the given value.

Variables

- `Ecore_List * ewl_embed_list`

8.75.1 Variable Documentation

8.75.1.1 `Ecore_List* ewl_embed_list`

8.76 lib/ewl_events.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Dnd_Types**
Provides type information about an external DND drag.
- struct **Ewl_Event_Action_Response**
Stores the reponse from a user action.
- struct **Ewl_Event_Dnd_Data_Received**
Provides information about dnd drop data.
- struct **Ewl_Event_Dnd_Data_Request**
Provides information about dnd data requests.
- struct **Ewl_Event_Dnd_Drop**
Provides information about dnd drops movement.
- struct **Ewl_Event_Dnd_Position**
Provides information about dnd drops movement.
- struct **Ewl_Event_Key**
Provides the basic information about key events.
- struct **Ewl_Event_Key_Down**
Provides clients with necessary information about the key press event.
- struct **Ewl_Event_Key_Up**
Provides clients with necessary information about the key release event.
- struct **Ewl_Event_Mouse**
Provides base mouse information.
- struct **Ewl_Event_Mouse_Down**
Provides information about the mouse down event.
- struct **Ewl_Event_Mouse_In**
Provides information about the mouse entering.
- struct **Ewl_Event_Mouse_Move**
Provides information about mouse movement.
- struct **Ewl_Event_Mouse_Out**

Provides information about the mouse leaving.

- struct **Ewl_Event_Mouse_Up**

Provides information about the mouse up event.

- struct **Ewl_Event_Mouse_Wheel**

Provides information about the mouse wheel scrolling.

- struct **Ewl_Event_Window_Expose**

Describes the canvas region that received an expose notification.

Typedefs

- typedef **Ewl_Dnd_Types** **Ewl_Dnd_Types**
- typedef **Ewl_Event_Action_Response** **Ewl_Event_Action_Response**
- typedef **Ewl_Event_Dnd_Data_Received** **Ewl_Event_Dnd_Data_Received**
- typedef **Ewl_Event_Dnd_Data_Request** **Ewl_Event_Dnd_Data_Request**
- typedef **Ewl_Event_Dnd_Drop** **Ewl_Event_Dnd_Drop**
- typedef **Ewl_Event_Dnd_Position** **Ewl_Event_Dnd_Position**
- typedef **Ewl_Event_Key** **Ewl_Event_Key**
- typedef **Ewl_Event_Key_Down** **Ewl_Event_Key_Down**
- typedef **Ewl_Event_Key_Up** **Ewl_Event_Key_Up**
- typedef **Ewl_Event_Mouse** **Ewl_Event_Mouse**
- typedef **Ewl_Event_Mouse_Down** **Ewl_Event_Mouse_Down**
- typedef **Ewl_Event_Mouse_In** **Ewl_Event_Mouse_In**
- typedef **Ewl_Event_Mouse_Move** **Ewl_Event_Mouse_Move**
- typedef **Ewl_Event_Mouse_Out** **Ewl_Event_Mouse_Out**
- typedef **Ewl_Event_Mouse_Up** **Ewl_Event_Mouse_Up**
- typedef **Ewl_Event_Mouse_Wheel** **Ewl_Event_Mouse_Wheel**
- typedef **Ewl_Event_Window_Expose** **Ewl_Event_Window_Expose**

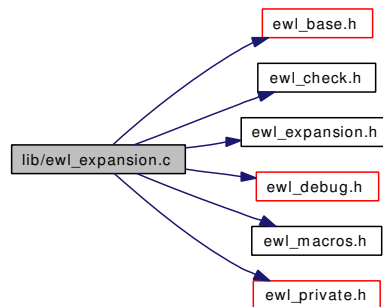
Functions

- unsigned int **ewl_ev_modifiers_get** (void)
Retrieve the current mask of modifiers that are set.
- void **ewl_ev_modifiers_set** (unsigned int modifiers)
Sets the key modifiers to the given value.

8.77 lib/ewl_expansion.c File Reference

```
#include "ewl_base.h"
#include "ewl_check.h"
#include "ewl_expansion.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_expansion.c:



Functions

- void **ewl_expansion_cb_reveal** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_expansion_cb_update_expandable** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_expansion_expandable_set** (Ewl_Expansion *cb, int c)
Change the expandable status of the expansion.
- void **ewl_expansion_expanded_set** (Ewl_Expansion *cb, int c)
Change the expanded status of the expansion.
- int **ewl_expansion_init** (Ewl_Expansion *cb)
Initialize the members and callbacks of a expansion.
- int **ewl_expansion_is_expandable** (Ewl_Expansion *cb)
Determine the expandable state of the expansion.
- int **ewl_expansion_is_expanded** (Ewl_Expansion *cb)
Determine the expanded state of the expansion.
- Ewl_Widget * **ewl_expansion_new** (void)
Allocate and initialize a new expansion.

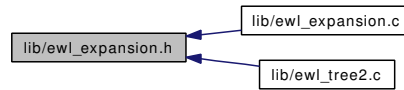
8.77.1 Function Documentation

8.77.1.1 void ewl_expansion_cb_reveal (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)

8.77.1.2 void ewl_expansion_cb_update_expandable (Ewl_Widget * *w*, void
*ev_data __ *UNUSED* __, void *user_data __ *UNUSED* __)

8.78 lib/ewl_expansion.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Expansion**

*Inherits from **Ewl_Check** (p. 597) and extends it to provide a expanded state.*

Defines

- `#define EWL_EXPANSION(c) ((Ewl_Expansion *)c)`
- `#define EWL_EXPANSION_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_EXPANSION_TYPE))`
- `#define EWL_EXPANSION_TYPE "expansion"`

Typedefs

- `typedef Ewl_Expansion Ewl_Expansion`

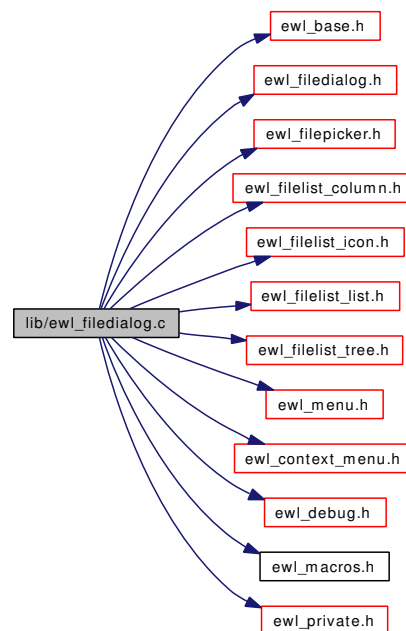
Functions

- `void ewl_expansion_cb_reveal (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_expansion_cb_update_expandable (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_expansion_expandable_set (Ewl_Expansion *c, int expandable)`
Change the expandable status of the expansion.
- `void ewl_expansion_expanded_set (Ewl_Expansion *c, int expanded)`
Change the expanded status of the expansion.
- `int ewl_expansion_init (Ewl_Expansion *c)`
Initialize the members and callbacks of a expansion.
- `int ewl_expansion_is_expandable (Ewl_Expansion *c)`
Determine the expandable state of the expansion.
- `int ewl_expansion_is_expanded (Ewl_Expansion *c)`
Determine the expanded state of the expansion.
- `Ewl_Widget * ewl_expansion_new (void)`
Allocate and initialize a new expansion.

8.79 lib/ewl_filedialog.c File Reference

```
#include "ewl_base.h"
#include "ewl_filedialog.h"
#include "ewl_filepicker.h"
#include "ewl_filelist_column.h"
#include "ewl_filelist_icon.h"
#include "ewl_filelist_list.h"
#include "ewl_filelist_tree.h"
#include "ewl_menu.h"
#include "ewl_context_menu.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_filedialog.c:



Functions

- void **ewl_filedialog_cb_delete_window** (Ewl_Widget *w, void *ev_data __UNUSED__, void *data __UNUSED__)
- const char * **ewl_filedialog_directory_get** (Ewl_Filedialog *fd)
Retrieve the current filedialog path.
- void **ewl_filedialog_directory_set** (Ewl_Filedialog *fd, const char *path)
Changes the current path of a filedialog.

- void **ewl_filedialog_filter_add** (**Ewl_Filedialog** *fd, const char *name, const char *filter)
Add the filter named name to the combo box in the filedialog.
- int **ewl_filedialog_init** (**Ewl_Filedialog** *fd)
Initialize a new filedialog.
- **Ewl_View** * **ewl_filedialog_list_view_get** (**Ewl_Filedialog** *fd)
Retrieve the file list view used in this file dialog.
- void **ewl_filedialog_list_view_set** (**Ewl_Filedialog** *fd, **Ewl_View** *view)
Set the view to be used for displaying the files in the dialog.
- unsigned int **ewl_filedialog_multiselect_get** (**Ewl_Filedialog** *fd)
gets the multiselect setting of the filedialog
- **Ewl_Widget** * **ewl_filedialog_multiselect_new** (void)
Create a new open filedialog.
- void **ewl_filedialog_multiselect_set** (**Ewl_Filedialog** *fd, unsigned int val)
Sets the dialog to multiselect or single select.
- **Ewl_Widget** * **ewl_filedialog_new** (void)
Create a new filedialog.
- char * **ewl_filedialog_selected_file_get** (**Ewl_Filedialog** *fd)
Retrieve the selected filename.
- void **ewl_filedialog_selected_file_set** (**Ewl_Filedialog** *fd, const char *file)
Set the currently selected file into the file dialog.
- **Ecore_List** * **ewl_filedialog_selected_files_get** (**Ewl_Filedialog** *fd)
returns all the elements selected by the user
- void **ewl_filedialog_selected_files_set** (**Ewl_Filedialog** *fd, **Ecore_List** *files)
Sets the given files as selected in the filedialog.
- unsigned int **ewl_filedialog_show_dot_files_get** (**Ewl_Filedialog** *fd)
Returns the current dot file setting of the dialog.
- void **ewl_filedialog_show_dot_files_set** (**Ewl_Filedialog** *fd, unsigned int dot)
Set if the file dialog should show dot files by default.
- unsigned int **ewl_filedialog_show_favorites_get** (**Ewl_Filedialog** *fd)
Get the current show favorites setting for the filedialog.
- void **ewl_filedialog_show_favorites_set** (**Ewl_Filedialog** *fd, unsigned int show)
Specify if the favorites column should be shown or not.

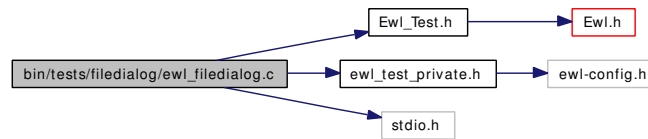
8.79.1 Function Documentation

8.79.1.1 void ewl_filedialog_cb_delete_window (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *data __ *UNUSED* __)

8.80 bin/tests/filedialog/ewl_filedialog.c File Reference

```
#include "Ewl_Test.h"  
#include "ewl_test_private.h"  
#include <stdio.h>
```

Include dependency graph for ewl_filedialog.c:



Functions

- `void test_info (Ewl_Test *test)`

8.80.1 Function Documentation

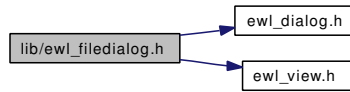
8.80.1.1 `void test_info (Ewl_Test * test)`

8.81 lib/ewl_filedialog.h File Reference

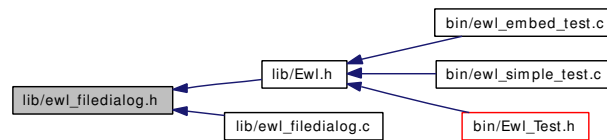
```
#include "ewl_dialog.h"
```

```
#include "ewl_view.h"
```

Include dependency graph for ewl_filedialog.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Filedialog**

*Inherits from **Ewl_Dialog** (p. 626) and extends to create a **filedialog**.*

Defines

- `#define EWL_FILEDIALOG(fd) ((Ewl_Filedialog *) fd)`
- `#define EWL_FILEDIALOG_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FILEDIALOG_TYPE))`
- `#define EWL_FILEDIALOG_TYPE "filedialog"`

Typedefs

- `typedef Ewl_Filedialog Ewl_Filedialog`

Functions

- `void ewl_filedialog_cb_delete_window (Ewl_Widget *w, void *ev_data, void *data)`
- `const char * ewl_filedialog_directory_get (Ewl_Filedialog *fd)`
Retrieve the current filedialog path.
- `void ewl_filedialog_directory_set (Ewl_Filedialog *fd, const char *dir)`
Changes the current path of a filedialog.
- `void ewl_filedialog_filter_add (Ewl_Filedialog *fd, const char *name, const char *filter)`

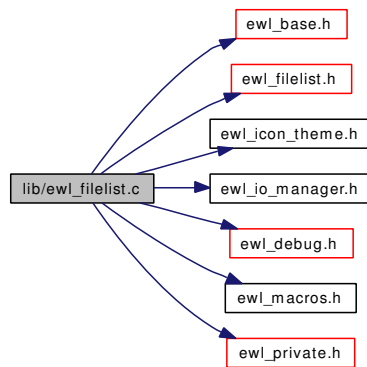
Add the filter named name to the combo box in the filedialog.

- **int ewl_filedialog_init (Ewl_Filedialog *fd)**
Initialize a new filedialog.
- **Ewl_View * ewl_filedialog_list_view_get (Ewl_Filedialog *fd)**
Retrieve the file list view used in this file dialog.
- **void ewl_filedialog_list_view_set (Ewl_Filedialog *fd, Ewl_View *view)**
Set the view to be used for displaying the files in the dialog.
- **unsigned int ewl_filedialog_multiselect_get (Ewl_Filedialog *fd)**
gets the multiselect setting of the filedialog
- **Ewl_Widget * ewl_filedialog_multiselect_new (void)**
Create a new open filedialog.
- **void ewl_filedialog_multiselect_set (Ewl_Filedialog *fd, unsigned int ms)**
Sets the dialog to multiselect or single select.
- **Ewl_Widget * ewl_filedialog_new (void)**
Create a new filedialog.
- **char * ewl_filedialog_selected_file_get (Ewl_Filedialog *fd)**
Retrieve the selected filename.
- **void ewl_filedialog_selected_file_set (Ewl_Filedialog *fd, const char *file)**
Set the currently selected file into the file dialog.
- **Ecore_List * ewl_filedialog_selected_files_get (Ewl_Filedialog *fd)**
returns all the elements selected by the user
- **void ewl_filedialog_selected_files_set (Ewl_Filedialog *fd, Ecore_List *files)**
Sets the given files as selected in the filedialog.
- **unsigned int ewl_filedialog_show_dot_files_get (Ewl_Filedialog *fd)**
Returns the current dot file setting of the dialog.
- **void ewl_filedialog_show_dot_files_set (Ewl_Filedialog *fd, unsigned int dot)**
Set if the file dialog should show dot files by default.
- **unsigned int ewl_filedialog_show_favorites_get (Ewl_Filedialog *fd)**
Get the current show favorites setting for the filedialog.
- **void ewl_filedialog_show_favorites_set (Ewl_Filedialog *fd, unsigned int show)**
Specify if the favorites column should be shown or not.

8.82 lib/ewl_filelist.c File Reference

```
#include "ewl_base.h"
#include "ewl_filelist.h"
#include "ewl_icon_theme.h"
#include "ewl_io_manager.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_filelist.c:



Functions

- void **ewl_filelist_cb_destroy** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_filelist_container_shift_handle** (Ewl_Filelist *fl, Ewl_Container *c, Ewl_Widget *clicked, const char *select_signal, const char *unselect_signal)
- const char * **ewl_filelist_directory_get** (Ewl_Filelist *fl)
Retrieves the current directory set on the filelist.
- void **ewl_filelist_directory_read** (Ewl_Filelist *fl, const char *dir, unsigned int skip_dot_dot, void(*func)(Ewl_Filelist *fl, const char *dir, char *file, void *data), void *data)
- void **ewl_filelist_directory_set** (Ewl_Filelist *fl, const char *dir)
Sets the given directory dir as the current directory in the filelist.
- char * **ewl_filelist_expand_path** (Ewl_Filelist *fl, const char *dir)
This will attempt to return the full path to the given directory. It should handle things like .. as well.
- const char * **ewl_filelist_filter_get** (Ewl_Filelist *fl)
Retrieves the current filter set on the filelist.
- void **ewl_filelist_filter_set** (Ewl_Filelist *fl, const char *filter)

Sets the given filter into the filelist.

- `char * ewl_filelist_groupname_get (gid_t st_gid)`
Converts the given group id into a group name.
- `void ewl_filelist_handle_click (Ewl_Filelist *fl, Ewl_Widget *w, Ewl_Event_Mouse_Up *ev, const char *select_state, const char *unselect_state)`
- `Ewl_Scrollpane_Flags ewl_filelist_hscroll_flag_get (Ewl_Filelist *fl)`
Retrieves the flags for the horizontal scrollbar.
- `void ewl_filelist_hscroll_flag_set (Ewl_Filelist *fl, Ewl_Scrollpane_Flags h)`
Sets the value to use for flags on the horizontal scrollbar.
- `int ewl_filelist_init (Ewl_Filelist *fl)`
Initialzie a filelist to default values.
- `char * ewl_filelist_modtime_get (time_t st_modtime)`
Converts the given modtime to a human readable string.
- `unsigned int ewl_filelist_multiselect_get (Ewl_Filelist *fl)`
Retrieves the current multiselect state of the filelist.
- `void ewl_filelist_multiselect_set (Ewl_Filelist *fl, unsigned int ms)`
Sets the given multiselect value into the filelist.
- `char * ewl_filelist_perms_get (mode_t st_mode)`
Converts the given mode settings into a human readable string.
- `char * ewl_filelist_selected_file_get (Ewl_Filelist *fl)`
Returns the currently selected file from the filelist.
- `Ewl_Widget * ewl_filelist_selected_file_preview_get (Ewl_Filelist *fl, const char *path)`
Creates and returns a preview widget for the given file.
- `void ewl_filelist_selected_file_set (Ewl_Filelist *fl, const char *file)`
Sets the given file as selected in the filelist.
- `void ewl_filelist_selected_files_change_notify (Ewl_Filelist *fl)`
Notifies interested consumers that the filelist has changed selected values.
- `Ecore_List * ewl_filelist_selected_files_get (Ewl_Filelist *fl)`
Retrieves the list of selected files in the filelist.
- `void ewl_filelist_selected_files_set (Ewl_Filelist *fl, Ecore_List *files)`
Sets the given files as selected in the filelist.
- `void ewl_filelist_selected_signal_all (Ewl_Filelist *fl, const char *signal)`
Signals all of the selected widgets with the given signal.

- unsigned int **ewl_filelist_show_dot_files_get** (Ewl_Filelist *fl)
Retrieves the current show dot files setting for the filelist.
- void **ewl_filelist_show_dot_files_set** (Ewl_Filelist *fl, unsigned int dot)
Sets the show dot files setting to the given value.
- char * **ewl_filelist_size_get** (off_t st_size)
Converts the given size into a human readable format.
- const char * **ewl_filelist_stock_icon_get** (Ewl_Filelist *fl, const char *path)
Retrieves the stock icon for the given file.
- char * **ewl_filelist_username_get** (uid_t st_uid)
Convertes the given user id into the appropriate user name.
- Ewl_Scrollpane_Flags **ewl_filelist_vscroll_flag_get** (Ewl_Filelist *fl)
Retrieves the flags for the vertical scrollbar.
- void **ewl_filelist_vscroll_flag_set** (Ewl_Filelist *fl, Ewl_Scrollpane_Flags v)
Sets the value to use for flags on the vertical scrollbar.

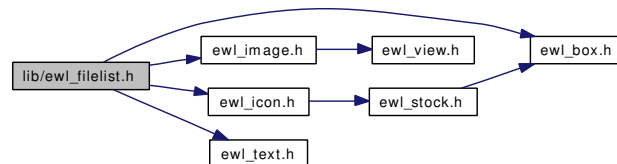
8.82.1 Function Documentation

8.82.1.1 void **ewl_filelist_cb_destroy** (Ewl_Widget * w, void *ev
__ *UNUSED* __, void *data __ *UNUSED* __)

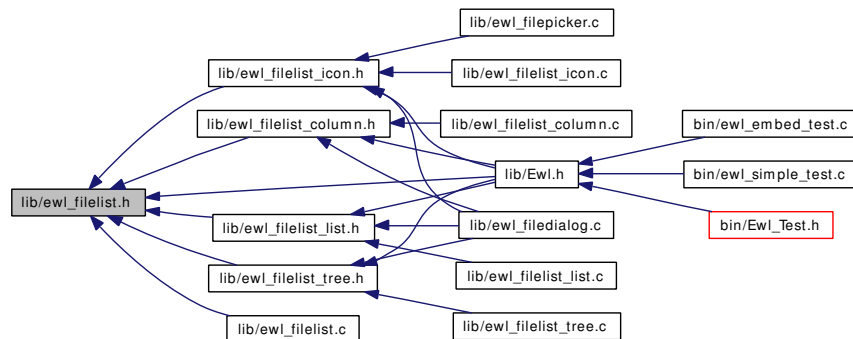
8.83 lib/ewl_filelist.h File Reference

```
#include "ewl_box.h"
#include "ewl_image.h"
#include "ewl_icon.h"
#include "ewl_text.h"
```

Include dependency graph for ewl_filelist.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Filelist**

*Inherits from **Ewl_Box** (p. 586) and extends to display a file list.*

Defines

- **#define EWL_FILELIST(f)** ((**Ewl_Filelist** *)f)
- **#define EWL_FILELIST_IS(w)** (ewl_widget_type_is(EWL_WIDGET(w), EWL_FILELIST_TYPE))
- **#define EWL_FILELIST_TYPE** "filelist"

Typedefs

- **typedef Ewl_Filelist Ewl_Filelist**

Functions

- void **ewl_filelist_cb_destroy** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_filelist_container_shift_handle** (Ewl_Filelist *fl, Ewl_Container *c, Ewl_Widget *clicked, const char *select_signal, const char *unselect_signal)
- const char * **ewl_filelist_directory_get** (Ewl_Filelist *fl)
Retrieves the current directory set on the filelist.
- void **ewl_filelist_directory_read** (Ewl_Filelist *fl, const char *dir, unsigned int skip_dot_dot, void(*func)(Ewl_Filelist *fl, const char *dir, char *file, void *data), void *data)
- void **ewl_filelist_directory_set** (Ewl_Filelist *fl, const char *dir)
Sets the given directory dir as the current directory in the filelist.
- char * **ewl_filelist_expand_path** (Ewl_Filelist *fl, const char *dir)
This will attempt to return the full path to the given directory. It should handle things like .. as well.
- const char * **ewl_filelist_filter_get** (Ewl_Filelist *fl)
Retrieves the current filter set on the filelist.
- void **ewl_filelist_filter_set** (Ewl_Filelist *fl, const char *filter)
Sets the given filter into the filelist.
- char * **ewl_filelist_groupname_get** (gid_t st_gid)
Converts the given group id into a group name.
- void **ewl_filelist_handle_click** (Ewl_Filelist *fl, Ewl_Widget *w, Ewl_Event_Mouse_Up *ev, const char *select_state, const char *unselect_state)
- Ewl_Scrollpane_Flags **ewl_filelist_hscroll_flag_get** (Ewl_Filelist *fl)
Retrieves the flags for the horizontal scrollbar.
- void **ewl_filelist_hscroll_flag_set** (Ewl_Filelist *fl, Ewl_Scrollpane_Flags h)
Sets the value to use for flags on the horizontal scrollbar.
- int **ewl_filelist_init** (Ewl_Filelist *fl)
Initialzie a filelist to default values.
- char * **ewl_filelist_modtime_get** (time_t modtime)
Converts the given modtime to a human readable string.
- unsigned int **ewl_filelist_multiselect_get** (Ewl_Filelist *fl)
Retrieves the current multiselect state of the filelist.
- void **ewl_filelist_multiselect_set** (Ewl_Filelist *fl, unsigned int ms)
Sets the given multiselect value into the filelist.
- char * **ewl_filelist_perms_get** (mode_t st_mode)
Converts the given mode settings into a human readable string.
- char * **ewl_filelist_selected_file_get** (Ewl_Filelist *fl)

Returns the currently selected file from the filelist.

- **Ewl_Widget * ewl_filelist_selected_file_preview_get** (Ewl_Filelist *fl, const char *path)

Creates and returns a preview widget for the given file.

- **void ewl_filelist_selected_file_set** (Ewl_Filelist *fl, const char *file)

Sets the given file as selected in the filelist.

- **void ewl_filelist_selected_files_change_notify** (Ewl_Filelist *fl)

Notifies interested consumers that the filelist has changed selected values.

- **Ecore_List * ewl_filelist_selected_files_get** (Ewl_Filelist *fl)

Retrieves the list of selected files in the filelist.

- **void ewl_filelist_selected_files_set** (Ewl_Filelist *fl, Ecore_List *files)

Sets the given files as selected in the filelist.

- **void ewl_filelist_selected_signal_all** (Ewl_Filelist *fl, const char *signal)

Signals all of the selected widgets with the given signal.

- **unsigned int ewl_filelist_show_dot_files_get** (Ewl_Filelist *fl)

Retrieves the current show dot files setting for the filelist.

- **void ewl_filelist_show_dot_files_set** (Ewl_Filelist *fl, unsigned int dot)

Sets the show dot files setting to the given value.

- **char * ewl_filelist_size_get** (off_t st_size)

Converts the given size into a human readable format.

- **const char * ewl_filelist_stock_icon_get** (Ewl_Filelist *fl, const char *path)

Retrieves the stock icon for the given file.

- **char * ewl_filelist_username_get** (uid_t st_uid)

Convertes the given user id into the appropriate user name.

- **Ewl_Scrollpane_Flags ewl_filelist_vscroll_flag_get** (Ewl_Filelist *fl)

Retrieves the flags for the vertical scrollbar.

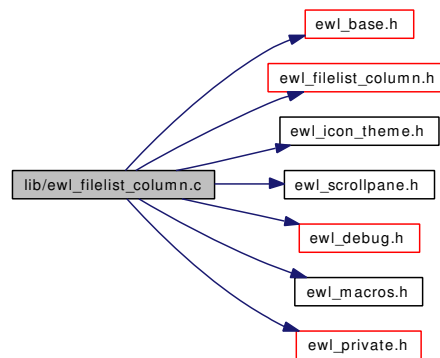
- **void ewl_filelist_vscroll_flag_set** (Ewl_Filelist *fl, Ewl_Scrollpane_Flags v)

Sets the value to use for flags on the vertical scrollbar.

8.84 lib/ewl_filelist_column.c File Reference

```
#include "ewl_base.h"
#include "ewl_filelist_column.h"
#include "ewl_icon_theme.h"
#include "ewl_scrollpane.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_filelist_column.c:



Data Structures

- struct **Ewl_Filelist_Column_Data**
Stores information on the filelist column data.

Typedefs

- typedef **Ewl_Filelist_Column_Data** **Ewl_Filelist_Column_Data**

Functions

- void **ewl_filelist_column_dir_change** (**Ewl_Filelist** *fl)
The callback for a directory change.
- const char * **ewl_filelist_column_filename_get** (**Ewl_Filelist** *fl, void *item)
Retrieves the filename for the given item.
- int **ewl_filelist_column_init** (**Ewl_Filelist_Column** *fl)
*Initializes an **Ewl_Filelist_Column** (p. 661) to default values.*
- **Ewl_Widget** * **ewl_filelist_column_new** (void)
*Creates a new **Ewl_Filelist_Column** (p. 661) widget.*

- void `ewl_filelist_column_selected_file_add` (`Ewl_Filelist *fl`, `const char *file`)
The callback to notify of selected files changing.
- void `ewl_filelist_column_selected_unselect` (`Ewl_Filelist *fl`)
- void `ewl_filelist_column_shift_handle` (`Ewl_Filelist *fl`, `Ewl_Widget *clicked`)
- `Ewl_View *ewl_filelist_column_view_get` (`void`)
*Retrieves the needed **Ewl_View** (p. 784) to use the column view.*

8.84.1 Typedef Documentation

8.84.1.1 typedef struct `Ewl_Filelist_Column_Data` `Ewl_Filelist_Column_Data`

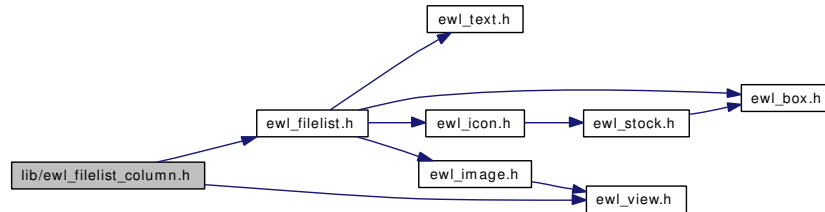
`Ewl_Filelist_Column_Data` (p. 663)

8.85 lib/ewl_filelist_column.h File Reference

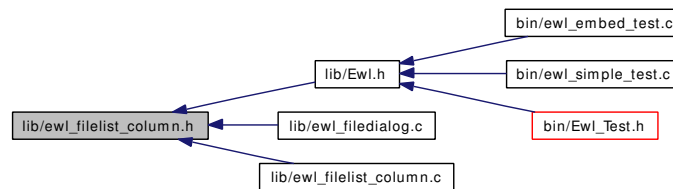
```
#include "ewl_filelist.h"
```

```
#include "ewl_view.h"
```

Include dependency graph for ewl_filelist_column.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Filelist_Column**

*Inherits from **Ewl_Filelist** (p. 658) and extends to provide a column view.*

Defines

- #define **EWL_FILELIST_COLUMN**(fl) ((**Ewl_Filelist_Column** *)fl)
- #define **EWL_FILELIST_COLUMN_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_FILELIST_COLUMN_TYPE**))
- #define **EWL_FILELIST_COLUMN_TYPE** "filelist_column"

Typedefs

- typedef **Ewl_Filelist_Column** **Ewl_Filelist_Column**

Functions

- void **ewl_filelist_column_dir_change** (**Ewl_Filelist** *fl)
The callback for a directory change.
- const char * **ewl_filelist_column_filename_get** (**Ewl_Filelist** *fl, void *item)

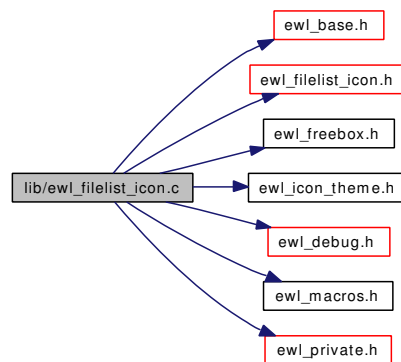
Retrieves the filename for the given item.

- `int ewl_filelist_column_init (Ewl_Filelist_Column *fl)`
*Initializes an **Ewl_Filelist_Column** (p. 661) to default values.*
- `Ewl_Widget * ewl_filelist_column_new (void)`
*Creates a new **Ewl_Filelist_Column** (p. 661) widget.*
- `void ewl_filelist_column_selected_file_add (Ewl_Filelist *fl, const char *file)`
The callback to notify of selected files changing.
- `void ewl_filelist_column_selected_unselect (Ewl_Filelist *fl)`
- `void ewl_filelist_column_shift_handle (Ewl_Filelist *fl, Ewl_Widget *clicked)`
- `Ewl_View * ewl_filelist_column_view_get (void)`
*Retrieves the needed **Ewl_View** (p. 784) to use the column view.*

8.86 lib/ewl_filelist_icon.c File Reference

```
#include "ewl_base.h"
#include "ewl_filelist_icon.h"
#include "ewl_freebox.h"
#include "ewl_icon_theme.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_filelist_icon.c:



Functions

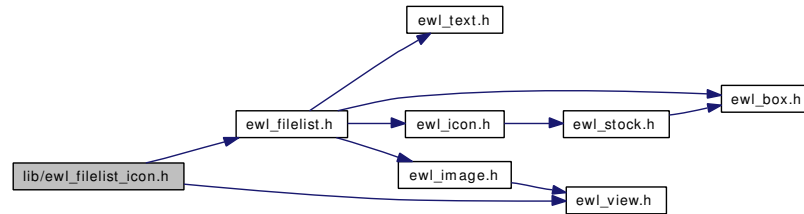
- void **ewl_filelist_icon_dir_change** (**Ewl_Filelist** *fl)
The callback for a directory change.
- const char * **ewl_filelist_icon_filename_get** (**Ewl_Filelist** *fl, void *item)
Retrieves the filename for the given item.
- int **ewl_filelist_icon_init** (**Ewl_Filelist_Icon** *fl)
*Initializes an **Ewl_Filelist_Icon** (p. 664) to default values.*
- **Ewl_Widget** * **ewl_filelist_icon_new** (void)
*Creates a new **Ewl_Filelist_Icon** (p. 664) widget.*
- void **ewl_filelist_icon_selected_file_add** (**Ewl_Filelist** *fl, const char *file)
The callback to notify of selected files changing.
- void **ewl_filelist_icon_selected_unselect** (**Ewl_Filelist** *fl)
- void **ewl_filelist_icon_shift_handle** (**Ewl_Filelist** *fl, **Ewl_Widget** *clicked)
- **Ewl_View** * **ewl_filelist_icon_view_get** (void)
*Retrieves the needed **Ewl_View** (p. 784) to use the icon view.*

8.87 lib/ewl_filelist_icon.h File Reference

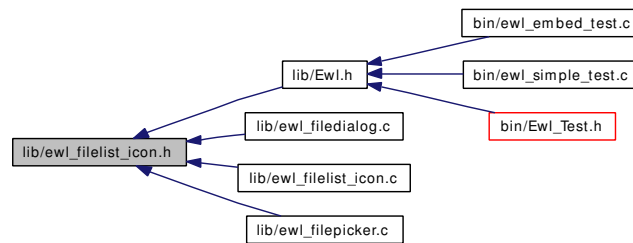
```
#include "ewl_filelist.h"
```

```
#include "ewl_view.h"
```

Include dependency graph for ewl_filelist_icon.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Filelist_Icon**

*Inherits from **Ewl_Filelist** (p. 658) and extends for an icon view.*

Defines

- #define **EWL_FILELIST_ICON**(fl) ((Ewl_Filelist_Icon *)fl)
- #define **EWL_FILELIST_ICON_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FILELIST_ICON_TYPE))
- #define **EWL_FILELIST_ICON_TYPE** "filelist_icon"

Typedefs

- typedef **Ewl_Filelist_Icon** **Ewl_Filelist_Icon**

Functions

- void **ewl_filelist_icon_dir_change** (Ewl_Filelist *fl)

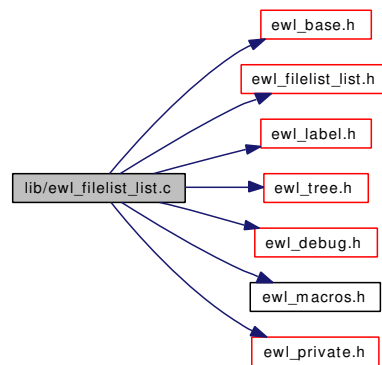
The callback for a directory change.

- `const char * ewl_filelist_icon_filename_get (Ewl_Filelist *fl, void *item)`
Retrieves the filename for the given item.
- `int ewl_filelist_icon_init (Ewl_Filelist_Icon *fl)`
*Initializes an **Ewl_Filelist_Icon** (p. 664) to default values.*
- `Ewl_Widget * ewl_filelist_icon_new (void)`
*Creates a new **Ewl_Filelist_Icon** (p. 664) widget.*
- `void ewl_filelist_icon_selected_file_add (Ewl_Filelist *fl, const char *file)`
The callback to notify of selected files changing.
- `void ewl_filelist_icon_selected_unselect (Ewl_Filelist *fl)`
- `void ewl_filelist_icon_shift_handle (Ewl_Filelist *fl, Ewl_Widget *clicked)`
- `Ewl_View * ewl_filelist_icon_view_get (void)`
*Retrieves the needed **Ewl_View** (p. 784) to use the icon view.*

8.88 lib/ewl_filelist_list.c File Reference

```
#include "ewl_base.h"
#include "ewl_filelist_list.h"
#include "ewl_label.h"
#include "ewl_tree.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_filelist_list.c:



Functions

- void **ewl_filelist_list_dir_change** (**Ewl_Filelist** *fl)
The callback to notify of a directory change.
- const char * **ewl_filelist_list_filename_get** (**Ewl_Filelist** *fl, void *item)
Retrieves the filename for the given item.
- int **ewl_filelist_list_init** (**Ewl_Filelist_List** *fl)
*Initializes an **Ewl_Filelist_List** (p. 665) widget to default values.*
- **Ewl_Widget** * **ewl_filelist_list_new** (void)
*Creates a new **Ewl_Filelist_List** (p. 665) widget.*
- void **ewl_filelist_list_selected_file_add** (**Ewl_Filelist** *fl, const char *file __-UNUSED__)
Callback when the selected files are changed.
- void **ewl_filelist_list_selected_unselect** (**Ewl_Filelist** *fl)
- void **ewl_filelist_list_shift_handle** (**Ewl_Filelist** *fl, **Ewl_Widget** *clicked)
- **Ewl_View** * **ewl_filelist_list_view_get** (void)
*Retrieves the **Ewl_View** (p. 784) needed to use the filelist list view.*

8.88.1 Function Documentation

8.88.1.1 void ewl_filelist_list_selected_file_add (Ewl_Filelist * *fl*, const char *file __ *UNUSED* __)

Callback when the selected files are changed.

Parameters:

fl,: The filelist to work with

file,: The file to set selected

Returns:

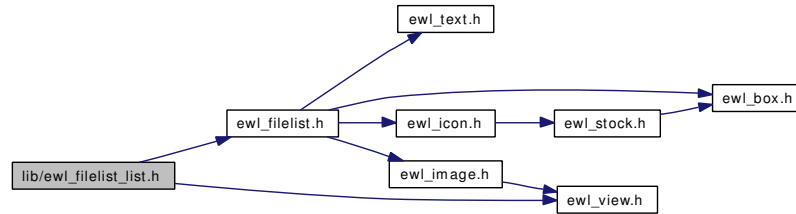
Returns no value

8.89 lib/ewl_filelist_list.h File Reference

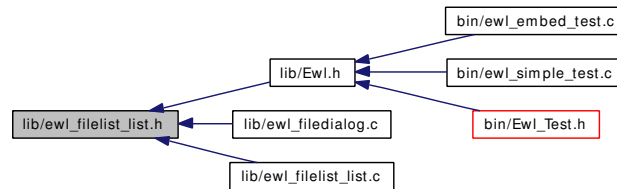
```
#include "ewl_filelist.h"
```

```
#include "ewl_view.h"
```

Include dependency graph for ewl_filelist_list.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Filelist_List**

*Inherits from **Ewl_Filelist** (p. 658) and extends to create a list view.*

Defines

- `#define EWL_FILELIST_LIST(fl) ((Ewl_Filelist_List *)fl)`
- `#define EWL_FILELIST_LIST_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FILELIST_LIST_TYPE))`
- `#define EWL_FILELIST_LIST_TYPE "filelist_list"`

Typedefs

- `typedef Ewl_Filelist_List Ewl_Filelist_List`

Functions

- `void ewl_filelist_list_dir_change (Ewl_Filelist *fl)`
The callback to notify of a directory change.
- `const char * ewl_filelist_list_filename_get (Ewl_Filelist *fl, void *item)`

Retrieves the filename for the given item.

- `int ewl_filelist_list_init (Ewl_Filelist_List *fl)`

*Initializes an **Ewl_Filelist_List** (p. 665) widget to default values.*

- `Ewl_Widget * ewl_filelist_list_new (void)`

*Creates a new **Ewl_Filelist_List** (p. 665) widget.*

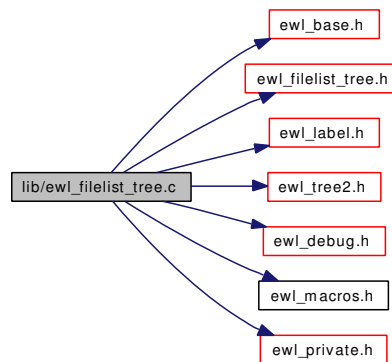
- `void ewl_filelist_list_selected_file_add (Ewl_Filelist *fl, const char *file)`
- `void ewl_filelist_list_selected_unselect (Ewl_Filelist *fl)`
- `void ewl_filelist_list_shift_handle (Ewl_Filelist *fl, Ewl_Widget *clicked)`
- `Ewl_View * ewl_filelist_list_view_get (void)`

*Retrieves the **Ewl_View** (p. 784) needed to use the filelist list view.*

8.90 lib/ewl_filelist_tree.c File Reference

```
#include "ewl_base.h"
#include "ewl_filelist_tree.h"
#include "ewl_label.h"
#include "ewl_tree2.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_filelist_tree.c:



Data Structures

- struct **Ewl_Filelist_Tree_Data**
Contains information on a filelist tree data.

Defines

- `#define EWL_FILELIST_TREE_DATA(data) ((Ewl_Filelist_Tree_Data*)(data))`

Typedefs

- `typedef Ewl_Filelist_Tree_Data Ewl_Filelist_Tree_Data`

Functions

- `void ewl_filelist_tree_dir_change (Ewl_Filelist *fl)`
The callback to notify of a directory change.
- `const char * ewl_filelist_tree_filename_get (Ewl_Filelist *fl, void *item)`
Retrieves the filename for the given item.

- `int ewl_filelist_tree_init (Ewl_Filelist_Tree *fl)`
*Initializes an **Ewl_Filelist_Tree** (p. 666) widget to default values.*
- `Ewl_Widget * ewl_filelist_tree_new (void)`
*Creates a new **Ewl_Filelist_Tree** (p. 666) widget.*
- `void ewl_filelist_tree_selected_file_add (Ewl_Filelist *fl, const char *file __-UNUSED__)`
Callback when the selected files are changed.
- `void ewl_filelist_tree_selected_unselect (Ewl_Filelist *fl)`
- `void ewl_filelist_tree_shift_handle (Ewl_Filelist *fl, Ewl_Widget *clicked)`
- `Ewl_View * ewl_filelist_tree_view_get (void)`
*Retrieves the **Ewl_View** (p. 784) needed to use the filelist tree view.*

8.90.1 Define Documentation

- 8.90.1.1 `#define EWL_FILELIST_TREE_DATA(data) ((Ewl_Filelist_Tree_Data *) (data))`

8.90.2 Typedef Documentation

- 8.90.2.1 `typedef struct Ewl_Filelist_Tree_Data Ewl_Filelist_Tree_Data`

`Ewl_Filelist_Tree_Data` (p. 667)

8.90.3 Function Documentation

- 8.90.3.1 `void ewl_filelist_tree_selected_file_add (Ewl_Filelist * fl, const char *file __-UNUSED__)`

Callback when the selected files are changed.

Parameters:

fl,: The filelist to work with

file,: The file to set selected

Returns:

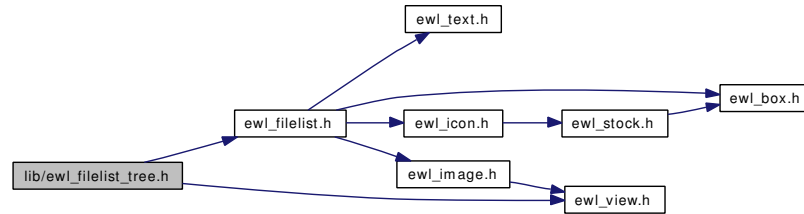
Returns no value

8.91 lib/ewl_filelist_tree.h File Reference

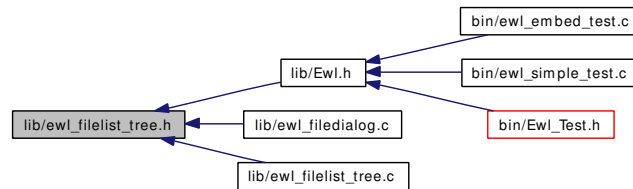
```
#include "ewl_filelist.h"
```

```
#include "ewl_view.h"
```

Include dependency graph for ewl_filelist_tree.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Filelist_Tree**

*Inherits from **Ewl_Filelist** (p. 658) and extends to provide a tree view.*

Defines

- #define **EWL_FILELIST_TREE**(fl) ((**Ewl_Filelist_Tree** *)fl)
- #define **EWL_FILELIST_TREE_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_FILELIST_TREE_TYPE**))
- #define **EWL_FILELIST_TREE_TYPE** "filelist_tree"

Typedefs

- typedef **Ewl_Filelist_Tree** **Ewl_Filelist_Tree**

Functions

- void **ewl_filelist_tree_dir_change** (**Ewl_Filelist** *fl)
The callback to notify of a directory change.
- const char * **ewl_filelist_tree_filename_get** (**Ewl_Filelist** *fl, void *item)

Retrieves the filename for the given item.

- `int ewl_filelist_tree_init (Ewl_Filelist_Tree *fl)`

*Initializes an **Ewl_Filelist_Tree** (p. 666) widget to default values.*

- `Ewl_Widget * ewl_filelist_tree_new (void)`

*Creates a new **Ewl_Filelist_Tree** (p. 666) widget.*

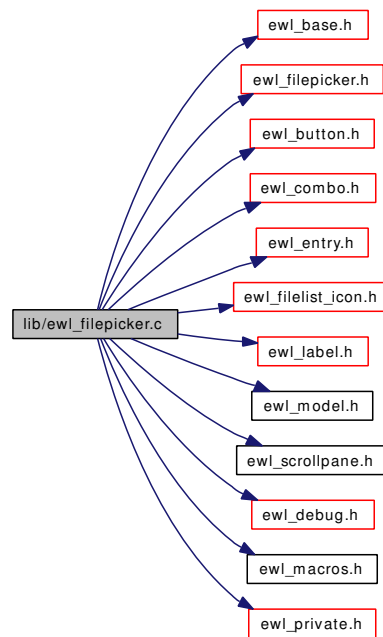
- `void ewl_filelist_tree_selected_file_add (Ewl_Filelist *fl, const char *file)`
- `void ewl_filelist_tree_selected_unselect (Ewl_Filelist *fl)`
- `void ewl_filelist_tree_shift_handle (Ewl_Filelist *fl, Ewl_Widget *clicked)`
- `Ewl_View * ewl_filelist_tree_view_get (void)`

*Retrieves the **Ewl_View** (p. 784) needed to use the filelist tree view.*

8.92 lib/ewl_filepicker.c File Reference

```
#include "ewl_base.h"
#include "ewl_filepicker.h"
#include "ewl_button.h"
#include "ewl_combo.h"
#include "ewl_entry.h"
#include "ewl_filelist_icon.h"
#include "ewl_label.h"
#include "ewl_model.h"
#include "ewl_scrollpane.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_filepicker.c:



Data Structures

- struct `Ewl_Filepicker_Filter`

Contains information on a filepicker filter.

Typedefs

- typedef **Ewl_Filepicker_Filter** **Ewl_Filepicker_Filter**

Functions

- const char * **ewl_filepicker_directory_get** (**Ewl_Filepicker** *fp)
Retrieve the current directory from the filepicker.
- void **ewl_filepicker_directory_set** (**Ewl_Filepicker** *fp, const char *dir)
Set the current directory to be displayed in the filepicker.
- void **ewl_filepicker_filter_add** (**Ewl_Filepicker** *fp, const char *name, const char *filter)
Add the filter named name to the combo box in the filepicker.
- const char * **ewl_filepicker_filter_get** (**Ewl_Filepicker** *fp)
Retrieves the current filter used in the filepicker.
- void **ewl_filepicker_filter_set** (**Ewl_Filepicker** *fp, const char *filter)
Set the given filter into the file picker.
- int **ewl_filepicker_init** (**Ewl_Filepicker** *fp)
*Initialize an **Ewl_Filepicker** (p. 668) to default values.*
- **Ewl_View** * **ewl_filepicker_list_view_get** (**Ewl_Filepicker** *fp)
Retrieves the current view used for the file list.
- void **ewl_filepicker_list_view_set** (**Ewl_Filepicker** *fp, **Ewl_View** *view)
Sets the given view to be used to display the file list.
- unsigned int **ewl_filepicker_multiselect_get** (**Ewl_Filepicker** *fp)
Retrieves the current multi-select value from the filepicker.
- void **ewl_filepicker_multiselect_set** (**Ewl_Filepicker** *fp, unsigned int ms)
Sets the multiselect state of the filepicker to the given value.
- **Ewl_Widget** * **ewl_filepicker_new** (void)
*Creates a new **Ewl_Filepicker** (p. 668) widget.*
- char * **ewl_filepicker_selected_file_get** (**Ewl_Filepicker** *fp)
Retrieves the currently selected file from the filepicker.
- void **ewl_filepicker_selected_file_set** (**Ewl_Filepicker** *fp, const char *file)
Sets the given file as selected in the filepicker.
- **Ecore_List** * **ewl_filepicker_selected_files_get** (**Ewl_Filepicker** *fp)
Retrieves the list of selected files in the filepicker.
- void **ewl_filepicker_selected_files_set** (**Ewl_Filepicker** *fp, **Ecore_List** *files)

Sets the given files as selected in the filepicker.

- unsigned int **ewl_filepicker_show_dot_files_get** (**Ewl_Filepicker** *fp)
Retrieves the current show dot files setting of the filepicker.
- void **ewl_filepicker_show_dot_files_set** (**Ewl_Filepicker** *fp, unsigned int dot)
Set the show dot files setting of the filepicker to the given value.
- unsigned int **ewl_filepicker_show_favorites_get** (**Ewl_Filepicker** *fp)
Retrieves the current show favorites setting of the filepicker.
- void **ewl_filepicker_show_favorites_set** (**Ewl_Filepicker** *fp, unsigned int show)
Sets the show favorites setting of the file picker.

8.92.1 Typedef Documentation

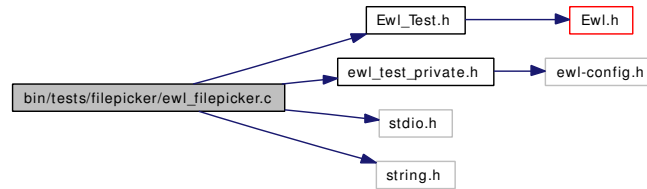
8.92.1.1 typedef struct Ewl_Filepicker_Filter Ewl_Filepicker_Filter

Ewl_Filepicker_Filter (p. 670)

8.93 bin/tests/filepicker/ewl_filepicker.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <string.h>
```

Include dependency graph for ewl_filepicker.c:



Functions

- `void test_info (Ewl_Test *test)`

8.93.1 Function Documentation

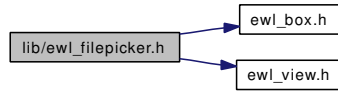
8.93.1.1 `void test_info (Ewl_Test * test)`

8.94 lib/ewl_filepicker.h File Reference

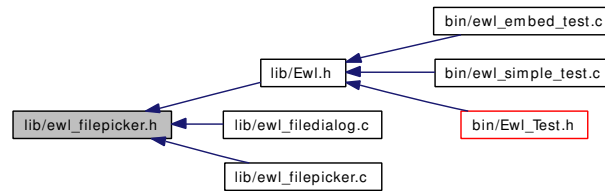
```
#include "ewl_box.h"
```

```
#include "ewl_view.h"
```

Include dependency graph for ewl_filepicker.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Filepicker**

*Inherits from **Ewl_Box** (p. 586) and extends to provide a filepicker.*

Defines

- `#define EWL_FILEPICKER(fp) ((Ewl_Filepicker *)fp)`
- `#define EWL_FILEPICKER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FILEPICKER_TYPE))`
- `#define EWL_FILEPICKER_TYPE "filepicker"`

Typedefs

- `typedef Ewl_Filepicker Ewl_Filepicker`

Functions

- `const char * ewl_filepicker_directory_get (Ewl_Filepicker *fp)`
Retrieve the current directory from the filepicker.
- `void ewl_filepicker_directory_set (Ewl_Filepicker *fp, const char *dir)`
Set the current directory to be displayed in the filepicker.
- `void ewl_filepicker_filter_add (Ewl_Filepicker *fp, const char *name, const char *filter)`

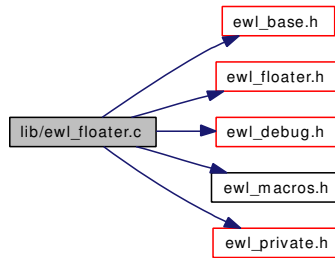
Add the filter named name to the combo box in the filepicker.

- `const char * ewl_filepicker_filter_get (Ewl_Filepicker *fp)`
Retrieves the current filter used in the filepicker.
- `void ewl_filepicker_filter_set (Ewl_Filepicker *fp, const char *filter)`
Set the given filter into the file picker.
- `int ewl_filepicker_init (Ewl_Filepicker *fp)`
*Initialize an **Ewl_Filepicker** (p. 668) to default values.*
- `Ewl_View * ewl_filepicker_list_view_get (Ewl_Filepicker *fp)`
Retrieves the current view used for the file list.
- `void ewl_filepicker_list_view_set (Ewl_Filepicker *fp, Ewl_View *view)`
Sets the given view to be used to display the file list.
- `unsigned int ewl_filepicker_multiselect_get (Ewl_Filepicker *fp)`
Retrieves the current multi-select value from the filepicker.
- `void ewl_filepicker_multiselect_set (Ewl_Filepicker *fp, unsigned int ms)`
Sets the multiselect state of the filepicker to the given value.
- `Ewl_Widget * ewl_filepicker_new (void)`
*Creates a new **Ewl_Filepicker** (p. 668) widget.*
- `char * ewl_filepicker_selected_file_get (Ewl_Filepicker *fp)`
Retrieves the currently selected file from the filepicker.
- `void ewl_filepicker_selected_file_set (Ewl_Filepicker *fp, const char *file)`
Sets the given file as selected in the filepicker.
- `Ecore_List * ewl_filepicker_selected_files_get (Ewl_Filepicker *fp)`
Retrieves the list of selected files in the filepicker.
- `void ewl_filepicker_selected_files_set (Ewl_Filepicker *fp, Ecore_List *files)`
Sets the given files as selected in the filepicker.
- `unsigned int ewl_filepicker_show_dot_files_get (Ewl_Filepicker *fp)`
Retrieves the current show dot files setting of the filepicker.
- `void ewl_filepicker_show_dot_files_set (Ewl_Filepicker *fp, unsigned int dot)`
Set the show dot files setting of the filepicker to the given value.
- `unsigned int ewl_filepicker_show_favorites_get (Ewl_Filepicker *fp)`
Retrieves the current show favorites setting of the filepicker.
- `void ewl_filepicker_show_favorites_set (Ewl_Filepicker *fp, unsigned int show)`
Sets the show favorites setting of the file picker.

8.95 lib/ewl_floater.c File Reference

```
#include "ewl_base.h"
#include "ewl_floater.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_floater.c:



Functions

- void **ewl_floater_cb_follow_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data)
- void **ewl_floater_cb_follow_destroy** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- Ewl_Widget * **ewl_floater_follow_get** (Ewl_Floater *f)

Get the widget we are following.

- void **ewl_floater_follow_set** (Ewl_Floater *f, Ewl_Widget *p)

Set the widget to follow.

- int **ewl_floater_init** (Ewl_Floater *f)

Initialize a floater to default values.

- Ewl_Widget * **ewl_floater_new** (void)

Allocate a new floater widget.

- void **ewl_floater_position_set** (Ewl_Floater *f, int x, int y)

Set the start x/y position of the floater.

- void **ewl_floater_relative_set** (Ewl_Floater *f, Ewl_Widget *w)

Set a floater positioned relative to a widget.

8.95.1 Function Documentation

8.95.1.1 void ewl_floater_cb_follow_configure (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void * *user_data*)

8.95.1.2 void ewl_floater_cb_follow_destroy (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)

8.95.1.3 void ewl_floater_relative_set (Ewl_Floater * *f*, Ewl_Widget * *w*)

Set a floater positioned relative to a widget.

Parameters:

f,: the floater to change positioning relation
w,: the widget to position the floater relative

Returns:

Returns no value.

Sets the floater *f* to be positioned relative to the position of the widget *w*.

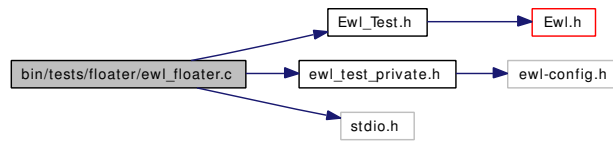
8.96 bin/tests/floater/ewl_floater.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_floater.c:



Functions

- `void test_info (Ewl_Test *test)`

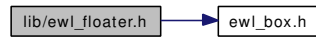
8.96.1 Function Documentation

8.96.1.1 `void test_info (Ewl_Test * test)`

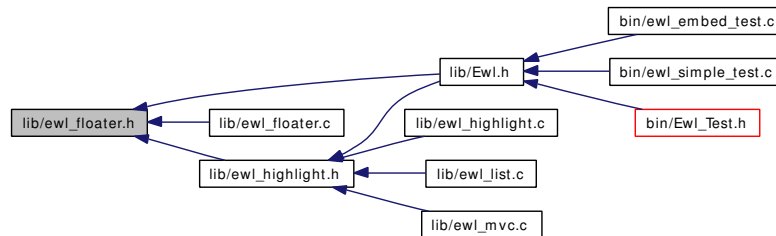
8.97 lib/ewl_floater.h File Reference

```
#include "ewl_box.h"
```

Include dependency graph for ewl_floater.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Floater**

*Inherits from **Ewl_Box** (p. 586) for layout purposes. Provides a means to layer above other widgets and to follow the movement of another widget.*

Defines

- `#define EWL_FLOATER(floater) ((Ewl_Floater *) floater)`
- `#define EWL_FLOATER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FLOATER_TYPE))`
- `#define EWL_FLOATER_TYPE "floater"`

Typedefs

- `typedef Ewl_Floater Ewl_Floater`

Functions

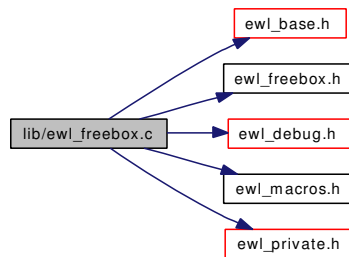
- `void ewl_floater_cb_follow_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_floater_cb_follow_destroy (Ewl_Widget *w, void *ev_data, void *user_data)`
- `Ewl_Widget * ewl_floater_follow_get (Ewl_Floater *f)`
Get the widget we are following.
- `void ewl_floater_follow_set (Ewl_Floater *f, Ewl_Widget *p)`
Set the widget to follow.

- `int ewl_floater_init (Ewl_Floater *f)`
Initialize a floater to default values.
- `Ewl_Widget * ewl_floater_new (void)`
Allocate a new floater widget.
- `void ewl_floater_position_set (Ewl_Floater *parent, int x, int y)`
Set the start x/y position of the floater.

8.98 lib/ewl_freebox.c File Reference

```
#include "ewl_base.h"
#include "ewl_freebox.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_freebox.c:



Functions

- void **ewl_freebox_cb_child_add** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_freebox_cb_child_show** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_freebox_cb_configure** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- Ewl_Freebox_Comparator **ewl_freebox_comparator_get** (Ewl_Freebox *fb)
*Retrieve the comparator set in the **Ewl_Freebox** (p. 673).*
- void **ewl_freebox_comparator_set** (Ewl_Freebox *fb, Ewl_Freebox_Comparator cmp)
*Set the comparator to use in the **Ewl_Freebox** (p. 673).*
- int **ewl_freebox_init** (Ewl_Freebox *fb)
*Initializes an **Ewl_Freebox** (p. 673) widget to default values.*
- Ewl_Freebox_Layout_Type **ewl_freebox_layout_type_get** (Ewl_Freebox *fb)
*Retrieve the layout type of the **Ewl_Freebox** (p. 673).*
- void **ewl_freebox_layout_type_set** (Ewl_Freebox *fb, Ewl_Freebox_Layout_Type type)
*Set the layout type of the **Ewl_Freebox** (p. 673).*
- Ewl_Widget * **ewl_freebox_new** (void)
*Creates a new, initialized **Ewl_Freebox** (p. 673) widget.*
- Ewl_Orientation **ewl_freebox_orientation_get** (Ewl_Freebox *fb)
Retrieve the current orientation of the freebox.

- void **ewl_freebox_orientation_set** (**Ewl_Freebox** *fb, **Ewl_Orientation** orientation)

Sets the orientation of the freebox.

- void **ewl_freebox_resort** (**Ewl_Freebox** *fb)

This will force the freebox to resort and redraw it's contents. Useful if part of the content changes but no child widgets where added/removed. (An icon label changes for example).

- **Ewl_Widget** * **ewl_hfreebox_new** (void)

creates and initializes a new freebox widget

- **Ewl_Widget** * **ewl_vfreebox_new** (void)

creates and initializes a new freebox widget

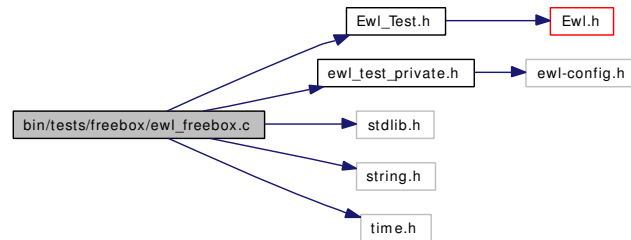
8.98.1 Function Documentation

8.98.1.1 void **ewl_freebox_cb_configure** (**Ewl_Widget** * w, void *ev
__ *UNUSED* __, void *data __ *UNUSED* __)

8.99 bin/tests/freebox/ewl_freebox.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdlib.h>
#include <string.h>
#include <time.h>
```

Include dependency graph for ewl_freebox.c:



Data Structures

- struct **Freebox__Test**

Typedefs

- typedef **Freebox__Test** **Freebox__Test**

Functions

- void **test__info** (**Ewl__Test** *test)

8.99.1 Typedef Documentation

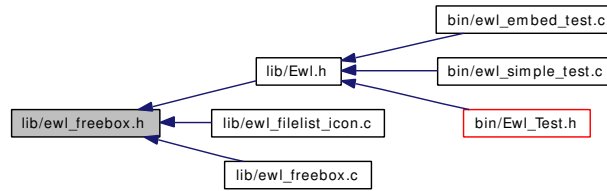
8.99.1.1 typedef struct Freebox__Test Freebox__Test

8.99.2 Function Documentation

8.99.2.1 void test__info (Ewl__Test * test)

8.100 lib/ewl_freebox.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Freebox**

*Inherits from **Ewl_Container** (p. 618) and extends to allow for differnt layout algorithms.*

Defines

- `#define EWL_FREEBOX(box) ((Ewl_Freebox *)box)`
- `#define EWL_FREEBOX_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_FREEBOX_TYPE))`
- `#define EWL_FREEBOX_TYPE "freebox"`

Typedefs

- `typedef Ewl_Freebox Ewl_Freebox`
- `typedef int(*) Ewl_Freebox_Comparator (Ewl_Widget *a, Ewl_Widget *b)`

Functions

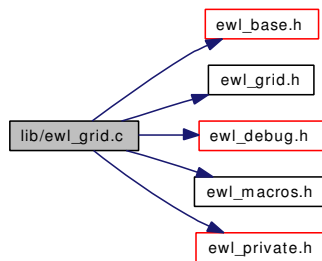
- `void ewl_freebox_cb_child_add (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_freebox_cb_child_show (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_freebox_cb_configure (Ewl_Widget *w, void *ev, void *data)`
- `Ewl_Freebox_Comparator ewl_freebox_comparator_get (Ewl_Freebox *fb)`
*Retrieve the comparator set in the **Ewl_Freebox** (p. 673).*
- `void ewl_freebox_comparator_set (Ewl_Freebox *fb, Ewl_Freebox_Comparator cmp)`
*Set the comparator to use in the **Ewl_Freebox** (p. 673).*
- `int ewl_freebox_init (Ewl_Freebox *fb)`
*Initializes an **Ewl_Freebox** (p. 673) widget to default values.*
- `Ewl_Freebox_Layout_Type ewl_freebox_layout_type_get (Ewl_Freebox *fb)`
*Retrieve the layout type of the **Ewl_Freebox** (p. 673).*

- void **ewl_freebox_layout_type_set** (**Ewl_Freebox** *fb, **Ewl_Freebox_Layout_Type** type)
*Set the layout type of the **Ewl_Freebox** (p. 673).*
- **Ewl_Widget** * **ewl_freebox_new** (void)
*Creates a new, initialized **Ewl_Freebox** (p. 673) widget.*
- **Ewl_Orientation** **ewl_freebox_orientation_get** (**Ewl_Freebox** *fb)
Retrieve the current orientation of the freebox.
- void **ewl_freebox_orientation_set** (**Ewl_Freebox** *fb, **Ewl_Orientation** orientation)
Sets the orientation of the freebox.
- void **ewl_freebox_resort** (**Ewl_Freebox** *fb)
This will force the freebox to resort and redraw it's contents. Useful if part of the content changes but no child widgets where added/removed. (An icon label changes for example).
- **Ewl_Widget** * **ewl_hfreebox_new** (void)
creates and initializes a new freebox widget
- **Ewl_Widget** * **ewl_vfreebox_new** (void)
creates and initializes a new freebox widget

8.101 lib/ewl_grid.c File Reference

```
#include "ewl_base.h"
#include "ewl_grid.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_grid.c:



Functions

- void ewl_grid_cb_child_add (Ewl_Container *p, Ewl_Widget *c)
- void ewl_grid_cb_child_remove (Ewl_Container *c, Ewl_Widget *w, int idx __UNUSED__)
- void ewl_grid_cb_child_resize (Ewl_Container *p, Ewl_Widget *child, int size __UNUSED__, Ewl_Orientation o __UNUSED__)
- void ewl_grid_cb_child_show (Ewl_Container *p, Ewl_Widget *child)
- void ewl_grid_cb_configure (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void ewl_grid_cb_destroy (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void ewl_grid_child_position_get (Ewl_Grid *g, Ewl_Widget *w, int *start_col, int *end_col, int *start_row, int *end_row)
get the position of a child widget
- void ewl_grid_child_position_set (Ewl_Grid *g, Ewl_Widget *w, int start_col, int end_col, int start_row, int end_row)
Give a child widget a fixed-postion in the grid.
- int ewl_grid_column_current_w_get (Ewl_Grid *g, int col)
Get the current width of a column.
- int ewl_grid_column_fixed_w_get (Ewl_Grid *g, int col)
Get the user set width of a column.
- void ewl_grid_column_fixed_w_set (Ewl_Grid *g, int col, int width)
Set the fixed size of a column.

- void **ewl_grid_column_preferred_w_use** (**Ewl_Grid** *g, int col)
use the preferred size of the column
- float **ewl_grid_column_relative_w_get** (**Ewl_Grid** *g, int col)
Get the user set relative width of a column.
- void **ewl_grid_column_relative_w_set** (**Ewl_Grid** *g, int col, float relw)
Set the relative width of a column.
- void **ewl_grid_column_w_remove** (**Ewl_Grid** *g, int col)
remove the user set size
- void **ewl_grid_dimensions_get** (**Ewl_Grid** *g, int *cols, int *rows)
get the number of columns and rows
- void **ewl_grid_dimensions_set** (**Ewl_Grid** *g, int cols, int rows)
sets the new dimensions
- unsigned int **ewl_grid_hhomogeneous_get** (**Ewl_Grid** *g)
Retrieves the horizontal homogeneous flag.
- void **ewl_grid_hhomogeneous_set** (**Ewl_Grid** *g, unsigned int h)
Change the horizontal homogeneous layout of the box.
- void **ewl_grid_homogeneous_set** (**Ewl_Grid** *g, unsigned int h)
Change the homogeneous layout of the grid.
- int **ewl_grid_init** (**Ewl_Grid** *g)
*Initializes an **Ewl_Grid** (p. 675) widget to default values.*
- **Ewl_Widget** * **ewl_grid_new** (void)
*Create a new **Ewl_Grid** (p. 675) widget.*
- **Ewl_Orientation** **ewl_grid_orientation_get** (**Ewl_Grid** *g)
Retrieves the fill orientation flag.
- void **ewl_grid_orientation_set** (**Ewl_Grid** *g, **Ewl_Orientation** orientation)
Change the fill orientation.
- int **ewl_grid_row_current_h_get** (**Ewl_Grid** *g, int row)
Get the current height of a column.
- int **ewl_grid_row_fixed_h_get** (**Ewl_Grid** *g, int row)
Get the user set height of a row.
- void **ewl_grid_row_fixed_h_set** (**Ewl_Grid** *g, int row, int height)
Set the fixed size of a column.
- void **ewl_grid_row_h_remove** (**Ewl_Grid** *g, int row)
remove the user set size

- void `ewl_grid_row_preferred_h_use` (`Ewl_Grid *g`, int row)
use the preferred size of the row
- float `ewl_grid_row_relative_h_get` (`Ewl_Grid *g`, int row)
Get the user set relative height of a row.
- void `ewl_grid_row_relative_h_set` (`Ewl_Grid *g`, int row, float relh)
Set the relative height of a row.
- unsigned int `ewl_grid_vhomogeneous_get` (`Ewl_Grid *g`)
Retrieves the vertical homogeneous flag.
- void `ewl_grid_vhomogeneous_set` (`Ewl_Grid *g`, unsigned int h)
Change the vertical homogeneous layout of the box.

8.101.1 Function Documentation

- 8.101.1.1 void `ewl_grid_cb_configure` (`Ewl_Widget * w`, void *`ev_data`
__ *UNUSED* __, void *`user_data` __ *UNUSED* __)
- 8.101.1.2 void `ewl_grid_cb_destroy` (`Ewl_Widget * w`, void *`ev_data`
__ *UNUSED* __, void *`user_data` __ *UNUSED* __)

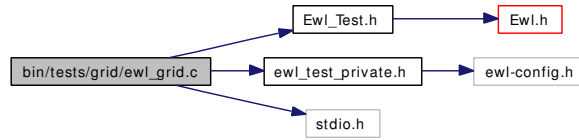
8.102 bin/tests/grid/ewl_grid.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_grid.c:



Functions

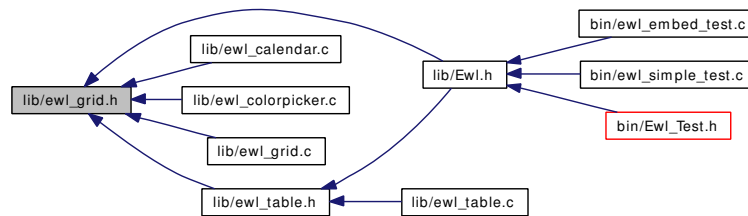
- `void test_info (Ewl_Test *test)`

8.102.1 Function Documentation

8.102.1.1 `void test_info (Ewl_Test * test)`

8.103 lib/ewl_grid.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Grid**
*Inherit from **Ewl_Container** (p. 618) and extend to provide a grid layout widget.*
- struct **Ewl_Grid_Child**
Contains information about a grid child.
- struct **Ewl_Grid_Info**
*Contains information about a row or column of **Ewl_Grid** (p. 675).*

Defines

- `#define EWL_GRID(grid) ((Ewl_Grid *)grid)`
- `#define EWL_GRID_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_GRID_TYPE))`
- `#define EWL_GRID_TYPE "grid"`

Typedefs

- `typedef Ewl_Grid Ewl_Grid`
- `typedef Ewl_Grid_Child Ewl_Grid_Child`
- `typedef Ewl_Grid_Info Ewl_Grid_Info`

Functions

- `void ewl_grid_cb_child_add (Ewl_Container *p, Ewl_Widget *c)`
- `void ewl_grid_cb_child_remove (Ewl_Container *p, Ewl_Widget *c, int idx)`
- `void ewl_grid_cb_child_resize (Ewl_Container *p, Ewl_Widget *child, int size, Ewl_Orientation o)`
- `void ewl_grid_cb_child_show (Ewl_Container *p, Ewl_Widget *child)`
- `void ewl_grid_cb_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_grid_cb_destroy (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_grid_child_position_get (Ewl_Grid *g, Ewl_Widget *child, int *start_col, int *end_col, int *start_row, int *end_row)`

get the position of a child widget

- void **ewl_grid_child_position_set** (**Ewl_Grid** *g, **Ewl_Widget** *child, int start_col, int end_col, int start_row, int end_row)

Give a child widget a fixed-position in the grid.

- int **ewl_grid_column_current_w_get** (**Ewl_Grid** *g, int col)

Get the current width of a column.

- int **ewl_grid_column_fixed_w_get** (**Ewl_Grid** *g, int col)

Get the user set width of a column.

- void **ewl_grid_column_fixed_w_set** (**Ewl_Grid** *g, int col, int width)

Set the fixed size of a column.

- void **ewl_grid_column_preferred_w_use** (**Ewl_Grid** *g, int col)

use the preferred size of the column

- float **ewl_grid_column_relative_w_get** (**Ewl_Grid** *g, int col)

Get the user set relative width of a column.

- void **ewl_grid_column_relative_w_set** (**Ewl_Grid** *g, int col, float relw)

Set the relative width of a column.

- void **ewl_grid_column_w_remove** (**Ewl_Grid** *g, int col)

remove the user set size

- void **ewl_grid_dimensions_get** (**Ewl_Grid** *g, int *col, int *row)

get the number of columns and rows

- void **ewl_grid_dimensions_set** (**Ewl_Grid** *g, int col, int row)

sets the new dimensions

- unsigned int **ewl_grid_hhomogeneous_get** (**Ewl_Grid** *g)

Retrieves the horizontal homogeneous flag.

- void **ewl_grid_hhomogeneous_set** (**Ewl_Grid** *g, unsigned int h)

Change the horizontal homogeneous layout of the box.

- void **ewl_grid_homogeneous_set** (**Ewl_Grid** *g, unsigned int h)

Change the homogeneous layout of the grid.

- int **ewl_grid_init** (**Ewl_Grid** *g)

*Initializes an **Ewl_Grid** (p. 675) widget to default values.*

- **Ewl_Widget** * **ewl_grid_new** (void)

*Create a new **Ewl_Grid** (p. 675) widget.*

- **Ewl_Orientation** **ewl_grid_orientation_get** (**Ewl_Grid** *g)

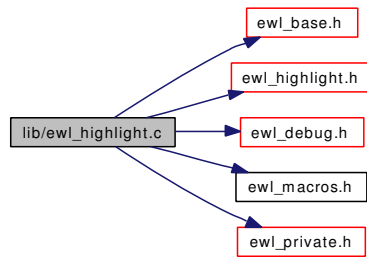
Retrieves the fill orientation flag.

- void **ewl_grid_orientation_set** (**Ewl_Grid** *g, **Ewl_Orientation** orientation)
Change the fill orientation.
- int **ewl_grid_row_current_h_get** (**Ewl_Grid** *g, int row)
Get the current height of a column.
- int **ewl_grid_row_fixed_h_get** (**Ewl_Grid** *g, int row)
Get the user set height of a row.
- void **ewl_grid_row_fixed_h_set** (**Ewl_Grid** *g, int row, int height)
Set the fixed size of a column.
- void **ewl_grid_row_h_remove** (**Ewl_Grid** *g, int row)
remove the user set size
- void **ewl_grid_row_preferred_h_use** (**Ewl_Grid** *g, int col)
use the preferred size of the row
- float **ewl_grid_row_relative_h_get** (**Ewl_Grid** *g, int col)
Get the user set relative height of a row.
- void **ewl_grid_row_relative_h_set** (**Ewl_Grid** *g, int col, float relh)
Set the relative height of a row.
- unsigned int **ewl_grid_vhomogeneous_get** (**Ewl_Grid** *g)
Retrieves the vertical homogeneous flag.
- void **ewl_grid_vhomogeneous_set** (**Ewl_Grid** *g, unsigned int h)
Change the vertical homogeneous layout of the box.

8.104 lib/ewl_highlight.c File Reference

```
#include "ewl_base.h"
#include "ewl_highlight.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_highlight.c:



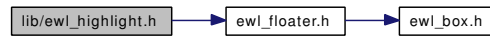
Functions

- **Ewl_Widget * ewl_highlight_follow_get (Ewl_Highlight *h)**
Retrieves the followed widget for the h highlight.
- **void ewl_highlight_follow_set (Ewl_Highlight *h, Ewl_Widget *w)**
Sets the highlight widget to follow the given w widget.
- **int ewl_highlight_init (Ewl_Highlight *h)**
*Initializes an **Ewl_Highlight** (p. 680) widget to default values.*
- **Ewl_Widget * ewl_highlight_new (void)**
Creates and returns a new widget on success or NULL on failure.

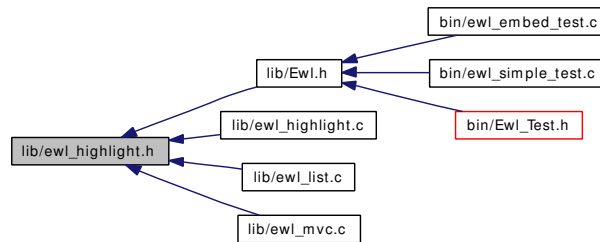
8.105 lib/ewl_highlight.h File Reference

```
#include "ewl_floater.h"
```

Include dependency graph for ewl_highlight.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Highlight**

*Inherits from **Ewl_Floater** (p. 671) and extends to provide a *highlight* widget.*

Defines

- `#define EWL_HIGHLIGHT(w) ((Ewl_Highlight *)w)`
- `#define EWL_HIGHLIGHT_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_HIGHLIGHT_TYPE))`
- `#define EWL_HIGHLIGHT_TYPE "highlight"`

Typedefs

- `typedef Ewl_Highlight Ewl_Highlight`

Functions

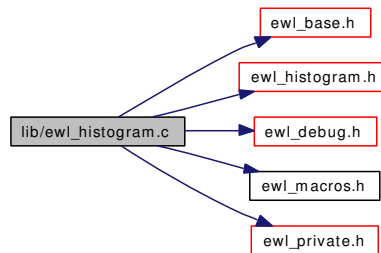
- `Ewl_Widget * ewl_highlight_follow_get (Ewl_Highlight *h)`
Retrieves the followed widget for the h highlight.
- `void ewl_highlight_follow_set (Ewl_Highlight *h, Ewl_Widget *w)`
Sets the highlight widget to follow the given w widget.
- `int ewl_highlight_init (Ewl_Highlight *h)`
*Initializes an **Ewl_Highlight** (p. 680) widget to default values.*
- `Ewl_Widget * ewl_highlight_new (void)`

Creates and returns a new widget on success or NULL on failure.

8.106 lib/ewl_histogram.c File Reference

```
#include "ewl_base.h"
#include "ewl_histogram.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_histogram.c:



Defines

- `#define A_CALC(color) ((color) >> 24)`
- `#define B_CALC(color) ((color << 24) >> 24)`
- `#define G_CALC(color) ((color << 16) >> 24)`
- `#define R_CALC(color) ((color << 8) >> 24)`
- `#define Y_CALC(color) (((R_CALC(color) * 299) + (G_CALC(color) * 587) + (B_CALC(color) * 114)) / 1000)`

Functions

- `void ewl_histogram_cb_configure (Ewl_Widget *w, void *event __UNUSED__, void *data __UNUSED__)`
- `Ewl_Histogram_Channel ewl_histogram_channel_get (Ewl_Histogram *hist)`
Sets the color channel to graph in the histogram.
- `void ewl_histogram_channel_set (Ewl_Histogram *hist, Ewl_Histogram_Channel channel)`
Sets the color channel to graph in the histogram.
- `void ewl_histogram_color_get (Ewl_Histogram *hist, int *r, int *g, int *b, int *a)`
Get the current color values for drawing the histogram.
- `void ewl_histogram_color_set (Ewl_Histogram *hist, int r, int g, int b, int a)`
Changes the drawing color of a histogram.
- `Ewl_Image * ewl_histogram_image_get (Ewl_Histogram *hist)`
Get the source image used to generate the histogram.

- void ewl_histogram_image_set (Ewl_Histogram *hist, Ewl_Image *image)
Change the source image used to generate the histogram.
- int ewl_histogram_init (Ewl_Histogram *hist)
Initialize a histogram widget to starting values.
- Ewl_Widget * ewl_histogram_new (void)
*Create a new **Ewl_Histogram** (p. 681) widget.*

8.106.1 Define Documentation

8.106.1.1 #define A_CALC(color) ((color) >> 24)

8.106.1.2 #define B_CALC(color) ((color << 24) >> 24)

8.106.1.3 #define G_CALC(color) ((color << 16) >> 24)

8.106.1.4 #define R_CALC(color) ((color << 8) >> 24)

8.106.1.5 #define Y_CALC(color) (((R_CALC(color) * 299) + (G_CALC(color) * 587) + (B_CALC(color) * 114)) / 1000)

8.106.2 Function Documentation

8.106.2.1 void ewl_histogram_cb_configure (Ewl_Widget * w, void *event __ *UNUSED* __, void *data __ *UNUSED* __)

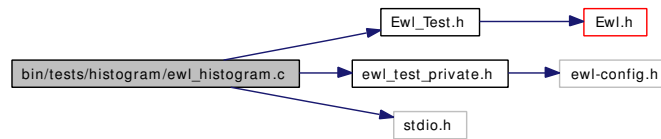
8.107 bin/tests/histogram/ewl_histogram.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_histogram.c:



Functions

- `void test_info (Ewl_Test *test)`

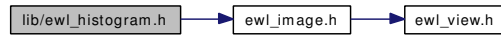
8.107.1 Function Documentation

8.107.1.1 `void test_info (Ewl_Test * test)`

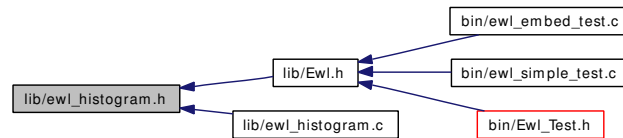
8.108 lib/ewl_histogram.h File Reference

```
#include "ewl_image.h"
```

Include dependency graph for ewl_histogram.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Histogram**

*Inherit from the **Ewl_Image** (p. 686) and extends to create a histogram widget.*

Defines

- #define **EWL_HISTOGRAM**(histogram) ((**Ewl_Histogram** *)histogram)
- #define **EWL_HISTOGRAM_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_HISTOGRAM_TYPE))
- #define **EWL_HISTOGRAM_TYPE** "histogram"

Typedefs

- typedef **Ewl_Histogram** **Ewl_Histogram**

Functions

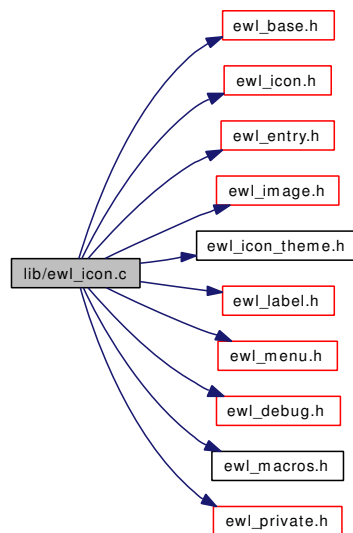
- void **ewl_histogram_cb_configure** (**Ewl_Widget** *w, void *event, void *data)
- **Ewl_Histogram_Channel** **ewl_histogram_channel_get** (**Ewl_Histogram** *histogram)
Sets the color channel to graph in the histogram.
- void **ewl_histogram_channel_set** (**Ewl_Histogram** *histogram, **Ewl_Histogram_Channel** channel)
Sets the color channel to graph in the histogram.
- void **ewl_histogram_color_get** (**Ewl_Histogram** *histogram, int *r, int *g, int *b, int *a)
Get the current color values for drawing the histogram.

- `void ewl_histogram_color_set (Ewl_Histogram *histogram, int r, int g, int b, int a)`
Changes the drawing color of a histogram.
- `Ewl_Image * ewl_histogram_image_get (Ewl_Histogram *histogram)`
Get the source image used to generate the histogram.
- `void ewl_histogram_image_set (Ewl_Histogram *histogram, Ewl_Image *image)`
Change the source image used to generate the histogram.
- `int ewl_histogram_init (Ewl_Histogram *histogram)`
Initialize a histogram widget to starting values.
- `Ewl_Widget * ewl_histogram_new (void)`
*Create a new **Ewl_Histogram** (p. 681) widget.*

8.109 lib/ewl_icon.c File Reference

```
#include "ewl_base.h"
#include "ewl_icon.h"
#include "ewl_entry.h"
#include "ewl_image.h"
#include "ewl_icon_theme.h"
#include "ewl_label.h"
#include "ewl_menu.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_icon.c:



Defines

- `#define EWL_ICON_COMPRESS_SIZE 10`

Functions

- `const char * ewl_icon_alt_text_get (Ewl_Icon *icon)`
Retrieves the alternate text set on the icon.
- `void ewl_icon_alt_text_set (Ewl_Icon *icon, const char *txt)`
Sets the given text as the alternate text for the icon.
- `void ewl_icon_cb_destroy (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)`

- unsigned int **ewl_icon_constrain_get** (**Ewl_Icon** *icon)
Retrieve the constrain value set on the icon.
- void **ewl_icon_constrain_set** (**Ewl_Icon** *icon, unsigned int val)
Set the constrain value on the icon.
- unsigned int **ewl_icon_editable_get** (**Ewl_Icon** *icon)
Retrieve if the icon is editable or not.
- void **ewl_icon_editable_set** (**Ewl_Icon** *icon, unsigned int e)
Set if the icon is editable or not.
- **Ewl_Widget** * **ewl_icon_extended_data_get** (**Ewl_Icon** *icon)
Retrieve the extended data from the icon.
- void **ewl_icon_extended_data_set** (**Ewl_Icon** *icon, **Ewl_Widget** *ext)
Set the extended data into the icon.
- const char * **ewl_icon_image_file_get** (**Ewl_Icon** *icon)
Retrieve the image to used in the icon.
- void **ewl_icon_image_set** (**Ewl_Icon** *icon, const char *file, const char *key)
set the image to use in the icon
- int **ewl_icon_init** (**Ewl_Icon** *icon)
*Initializes the given **Ewl_Icon** (p. 683) widget.*
- unsigned int **ewl_icon_label_complex_get** (**Ewl_Icon** *icon)
Retrieve if the icon has a complex label or not.
- void **ewl_icon_label_complex_set** (**Ewl_Icon** *icon, unsigned int c)
Set if the icon use a label or a text widget for the label part.
- unsigned int **ewl_icon_label_compressed_get** (**Ewl_Icon** *icon)
Retrieves the current compressiion setting for the icon.
- void **ewl_icon_label_compressed_set** (**Ewl_Icon** *icon, unsigned int compress)
Sets the compressions setting for the icon to the given value.
- const char * **ewl_icon_label_get** (**Ewl_Icon** *icon)
Retrieve the label from the icon.
- void **ewl_icon_label_set** (**Ewl_Icon** *icon, const char *label)
Set the label of the icon.
- **Ewl_Widget** * **ewl_icon_menu_get** (**Ewl_Icon** *icon)
Retrieve the menu from the icon.
- void **ewl_icon_menu_set** (**Ewl_Icon** *icon, **Ewl_Widget** *menu)
Set the menu for the icon.

- **Ewl_Widget * ewl_icon_new** (void)
*Creates and initializes a new **Ewl_Icon** (p. 683) widget.*
- **void ewl_icon_part_hide** (Ewl_Icon *icon, Ewl_Icon_Part part)
Hide the given part of the icon. Note: You can only hide one part.
- **Ewl_Widget * ewl_icon_simple_new** (void)
*Creates and initializes a new **Ewl_Icon** (p. 683) widget.*
- **unsigned int ewl_icon_thumbnailing_get** (Ewl_Icon *icon)
Retrieve if the icon use thumbnailing.
- **void ewl_icon_thumbnailing_set** (Ewl_Icon *icon, unsigned int t)
Set if the icon thumbnailed the image or use it directly.
- **Ewl_Icon_Type ewl_icon_type_get** (Ewl_Icon *icon)
Retrieve the type of the icon.
- **void ewl_icon_type_set** (Ewl_Icon *icon, Ewl_Icon_Type type)
Set the type of the icon.

8.109.1 Define Documentation

8.109.1.1 **#define EWL_ICON_COMPRESS_SIZE 10**

8.109.2 Function Documentation

8.109.2.1 **void ewl_icon_cb_destroy** (Ewl_Widget * w, void *ev __ *UNUSED* __, void *data __ *UNUSED* __)

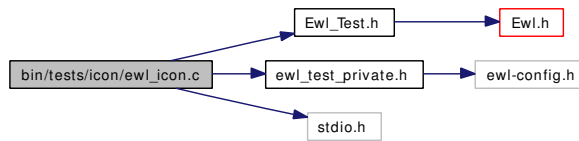
8.110 bin/tests/icon/ewl_icon.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_icon.c:



Functions

- `void test_info (Ewl_Test *test)`

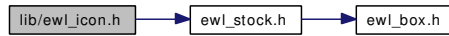
8.110.1 Function Documentation

8.110.1.1 `void test_info (Ewl_Test * test)`

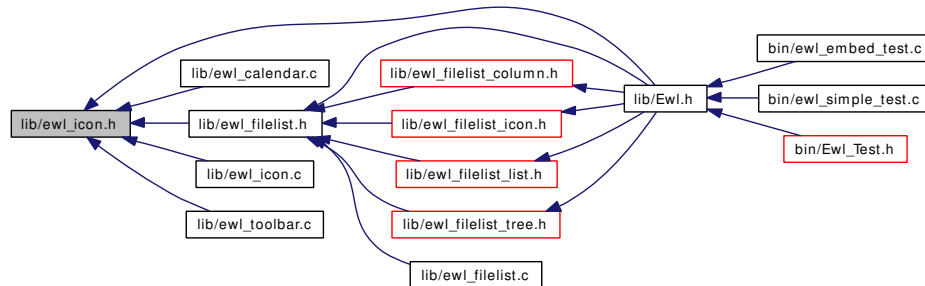
8.111 lib/ewl_icon.h File Reference

```
#include "ewl_stock.h"
```

Include dependency graph for ewl_icon.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Icon**

*Inherits from **Ewl_Stock** (p. 750) and extends to provide for an icon layout.*

Defines

- `#define EWL_ICON(icon) ((Ewl_Icon *)icon)`
- `#define EWL_ICON_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_ICON_TYPE))`
- `#define EWL_ICON_TYPE "icon"`

Typedefs

- `typedef Ewl_Icon Ewl_Icon`

Functions

- `const char * ewl_icon_alt_text_get (Ewl_Icon *icon)`
Retrieves the alternate text set on the icon.
- `void ewl_icon_alt_text_set (Ewl_Icon *icon, const char *txt)`
Sets the given text as the alternate text for the icon.
- `void ewl_icon_cb_destroy (Ewl_Widget *w, void *ev, void *data)`

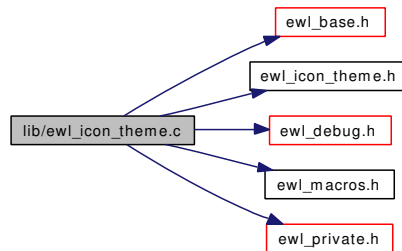
- unsigned int **ewl_icon_constrain_get** (**Ewl_Icon** *icon)
Retrieve the constrain value set on the icon.
- void **ewl_icon_constrain_set** (**Ewl_Icon** *icon, unsigned int val)
Set the constrain value on the icon.
- unsigned int **ewl_icon_editable_get** (**Ewl_Icon** *icon)
Retrieve if the icon is editable or not.
- void **ewl_icon_editable_set** (**Ewl_Icon** *icon, unsigned int e)
Set if the icon is editable or not.
- **Ewl_Widget** * **ewl_icon_extended_data_get** (**Ewl_Icon** *icon)
Retrieve the extended data from the icon.
- void **ewl_icon_extended_data_set** (**Ewl_Icon** *icon, **Ewl_Widget** *ext)
Set the extended data into the icon.
- const char * **ewl_icon_image_file_get** (**Ewl_Icon** *icon)
Retrieve the image to used in the icon.
- void **ewl_icon_image_set** (**Ewl_Icon** *icon, const char *file, const char *key)
set the image to use in the icon
- int **ewl_icon_init** (**Ewl_Icon** *icon)
*Initializes the given **Ewl_Icon** (p. 683) widget.*
- unsigned int **ewl_icon_label_complex_get** (**Ewl_Icon** *icon)
Retrieve if the icon has a complex label or not.
- void **ewl_icon_label_complex_set** (**Ewl_Icon** *icon, unsigned int complex_label)
Set if the icon use a label or a text widget for the label part.
- unsigned int **ewl_icon_label_compressed_get** (**Ewl_Icon** *icon)
Retrieves the current compressiion setting for the icon.
- void **ewl_icon_label_compressed_set** (**Ewl_Icon** *icon, unsigned int compress)
Sets the compressions setting for the icon to the given value.
- const char * **ewl_icon_label_get** (**Ewl_Icon** *icon)
Retrieve the label from the icon.
- void **ewl_icon_label_set** (**Ewl_Icon** *icon, const char *label)
Set the label of the icon.
- **Ewl_Widget** * **ewl_icon_menu_get** (**Ewl_Icon** *icon)
Retrieve the menu from the icon.
- void **ewl_icon_menu_set** (**Ewl_Icon** *icon, **Ewl_Widget** *menu)
Set the menu for the icon.

- **Ewl_Widget * ewl_icon_new** (void)
*Creates and initializes a new **Ewl_Icon** (p. 683) widget.*
- **void ewl_icon_part_hide** (Ewl_Icon *icon, Ewl_Icon_Part part)
Hide the given part of the icon. Note: You can only hide one part.
- **Ewl_Widget * ewl_icon_simple_new** (void)
*Creates and initializes a new **Ewl_Icon** (p. 683) widget.*
- **unsigned int ewl_icon_thumbnailing_get** (Ewl_Icon *icon)
Retrieve if the icon use thumbnailing.
- **void ewl_icon_thumbnailing_set** (Ewl_Icon *icon, unsigned int thumb)
Set if the icon thumbnailed the image or use it directly.
- **Ewl_Icon_Type ewl_icon_type_get** (Ewl_Icon *icon)
Retrieve the type of the icon.
- **void ewl_icon_type_set** (Ewl_Icon *icon, Ewl_Icon_Type type)
Set the type of the icon.

8.112 lib/ewl_icon_theme.c File Reference

```
#include "ewl_base.h"
#include "ewl_icon_theme.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_icon_theme.c:



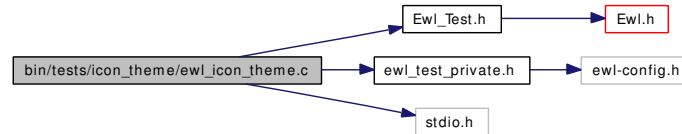
Functions

- `const char * ewl_icon_theme_icon_path_get (const char *icon, int size)`
Retrives the full path to the specified icon, or NULL if none found.
- `int ewl_icon_theme_init (void)`
Initializes the icon theme system.
- `void ewl_icon_theme_shutdown (void)`
Shuts down the icon theme system.
- `void ewl_icon_theme_theme_change (void)`
Called when the icon theme is changed so we can clean up any caching we have in place.

8.113 bin/tests/icon_theme/ewl_icon_theme.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
```

Include dependency graph for ewl_icon_theme.c:



Defines

- `#define EWL_ICON_THEME_PER_LOOP 20`

Functions

- `void test_info (Ewl_Test *test)`

Variables

- `const char * icons []`

8.113.1 Define Documentation

8.113.1.1 `#define EWL_ICON_THEME_PER_LOOP 20`

8.113.2 Function Documentation

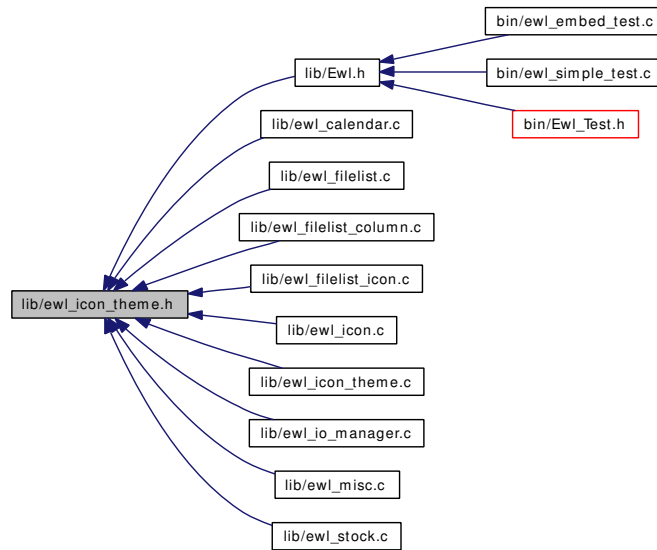
8.113.2.1 `void test_info (Ewl_Test * test)`

8.113.3 Variable Documentation

8.113.3.1 `const char* icons[]`

8.114 lib/ewl_icon_theme.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define EWL_ICON_ACCESSORIES_CALCULATOR` "accessories-calculator"
- `#define EWL_ICON_ACCESSORIES_CHARACTER_MAP` "accessories-character-map"
- `#define EWL_ICON_ACCESSORIES_DICTIONARY` "accessories-dictionary"
- `#define EWL_ICON_ACCESSORIES_TEXT_EDITOR` "accessories-text-editor"
- `#define EWL_ICON_ADDRESS_BOOK_NEW` "address-book-new"
- `#define EWL_ICON_APPLICATION_EXIT` "application-exit"
- `#define EWL_ICON_APPLICATION_X_EXECUTABLE` "application-x-executable"
- `#define EWL_ICON_APPLICATIONS_ACCESSORIES` "applications-accessories"
- `#define EWL_ICON_APPLICATIONS_DEVELOPMENT` "applications-development"
- `#define EWL_ICON_APPLICATIONS_GAMES` "applications-games"
- `#define EWL_ICON_APPLICATIONS_GRAPHICS` "applications-graphics"
- `#define EWL_ICON_APPLICATIONS_INTERNET` "applications-internet"
- `#define EWL_ICON_APPLICATIONS_MULTIMEDIA` "applications-multimedia"
- `#define EWL_ICON_APPLICATIONS_OFFICE` "applications-office"
- `#define EWL_ICON_APPLICATIONS_OTHER` "applications-other"
- `#define EWL_ICON_APPLICATIONS_SYSTEM` "applications-system"
- `#define EWL_ICON_APPLICATIONS_UTILITIES` "applications-utilities"
- `#define EWL_ICON_APPOINTMENT_MISSED` "appointment-missed"
- `#define EWL_ICON_APPOINTMENT_NEW` "appointment-new"
- `#define EWL_ICON_APPOINTMENT_SOON` "appointment-soon"
- `#define EWL_ICON_AUDIO_CARD` "audio-card"

- #define EWL_ICON_AUDIO_INPUT_MICROPHONE "audio-input-microphone"
- #define EWL_ICON_AUDIO_VOLUME_HIGH "audio-volume-high"
- #define EWL_ICON_AUDIO_VOLUME_LOW "audio-volume-low"
- #define EWL_ICON_AUDIO_VOLUME_MEDIUM "audio-volume-medium"
- #define EWL_ICON_AUDIO_VOLUME_MUTED "audio-volume-muted"
- #define EWL_ICON_AUDIO_X_GENERIC "audio-x-generic"
- #define EWL_ICON_BATTERY "battery"
- #define EWL_ICON_BATTERY_CAUTION "battery-caution"
- #define EWL_ICON_BATTERY_LOW "battery-low"
- #define EWL_ICON_CAMERA_PHOTO "camera-photo"
- #define EWL_ICON_CAMERA_VIDEO "camera-video"
- #define EWL_ICON_COMPUTER "computer"
- #define EWL_ICON_CONTACT_NEW "contact-new"
- #define EWL_ICON_DIALOG_APPLY "dialog-apply"
- #define EWL_ICON_DIALOG_CANCEL "dialog-cancel"
- #define EWL_ICON_DIALOG_CLOSE "dialog-close"
- #define EWL_ICON_DIALOG_ERROR "dialog-error"
- #define EWL_ICON_DIALOG_INFORMATION "dialog-information"
- #define EWL_ICON_DIALOG_OK "dialog-ok"
- #define EWL_ICON_DIALOG_PASSWORD "dialog-password"
- #define EWL_ICON_DIALOG_QUESTION "dialog-question"
- #define EWL_ICON_DIALOG_WARNING "dialog-warning"
- #define EWL_ICON_DOCUMENT_NEW "document-new"
- #define EWL_ICON_DOCUMENT_OPEN "document-open"
- #define EWL_ICON_DOCUMENT_OPEN_RECENT "document-open-recent"
- #define EWL_ICON_DOCUMENT_PAGE_SETUP "document-page-setup"
- #define EWL_ICON_DOCUMENT_PRINT "document-print"
- #define EWL_ICON_DOCUMENT_PRINT_PREVIEW "document-print-preview"
- #define EWL_ICON_DOCUMENT_PROPERTIES "document-properties"
- #define EWL_ICON_DOCUMENT_REVERT "document-revert"
- #define EWL_ICON_DOCUMENT_SAVE "document-save"
- #define EWL_ICON_DOCUMENT_SAVE_AS "document-save-as"
- #define EWL_ICON_DRIVE_CDROM "drive-cdrom"
- #define EWL_ICON_DRIVE_HARDDISK "drive-harddisk"
- #define EWL_ICON_DRIVE_REMOVABLE_MEDIA "drive-removable-media"
- #define EWL_ICON_EDIT_COPY "edit-copy"
- #define EWL_ICON_EDIT_CUT "edit-cut"
- #define EWL_ICON_EDIT_DELETE "edit-delete"
- #define EWL_ICON_EDIT_FIND "edit-find"
- #define EWL_ICON_EDIT_FIND_REPLACE "edit-find-replace"
- #define EWL_ICON_EDIT_PASTE "edit-paste"
- #define EWL_ICON_EDIT_REDO "edit-redo"
- #define EWL_ICON_EDIT_SELECT_ALL "edit-select-all"
- #define EWL_ICON_EDIT_UNDO "edit-undo"
- #define EWL_ICON_EMBLEM_DEFAULT "emblem-default"
- #define EWL_ICON_EMBLEM_DOCUMENTS "emblem-documents"
- #define EWL_ICON_EMBLEM_DOWNLOADS "emblem-downloads"
- #define EWL_ICON_EMBLEM_FAVORITE "emblem-favorite"

```

• #define EWL_ICON_EMBLEM_IMPORTANT "emblem-important"
• #define EWL_ICON_EMBLEM_MAIL "emblem-mail"
• #define EWL_ICON_EMBLEM_PHOTOS "emblem-photos"
• #define EWL_ICON_EMBLEM_READONLY "emblem-readonly"
• #define EWL_ICON_EMBLEM_SHARED "emblem-shared"
• #define EWL_ICON_EMBLEM_SYMBOLIC_LINK "emblem-symbolic-link"
• #define EWL_ICON_EMBLEM_SYNCHRONIZED "emblem-synchronized"
• #define EWL_ICON_EMBLEM_SYSTEM "emblem-system"
• #define EWL_ICON_EMBLEM_UNREADABLE "emblem-unreadable"
• #define EWL_ICON_FACE_ANGEL "face-angel"
• #define EWL_ICON_FACE_CRYING "face-crying"
• #define EWL_ICON_FACE_DEVIL_GRIN "face-devil-grin"
• #define EWL_ICON_FACE_DEVIL_SAD "face-devil-sad"
• #define EWL_ICON_FACE_GLASSES "face-glasses"
• #define EWL_ICON_FACE_KISS "face-kiss"
• #define EWL_ICON_FACE_MONKEY "face-monkey"
• #define EWL_ICON_FACE_PLAIN "face-plain"
• #define EWL_ICON_FACE_SAD "face-sad"
• #define EWL_ICON_FACE_SMILE "face-smile"
• #define EWL_ICON_FACE_SMILE_BIG "face-smile-big"
• #define EWL_ICON_FACE_SMIRK "face-smirk"
• #define EWL_ICON_FACE_SURPRISE "face-surprise"
• #define EWL_ICON_FACE_WINK "face-wink"
• #define EWL_ICON_FOLDER "folder"
• #define EWL_ICON_FOLDER_DRAG_ACCEPT "folder-drag-accept"
• #define EWL_ICON_FOLDER_OPEN "folder-open"
• #define EWL_ICON_FOLDER_REMOTE "folder-remote"
• #define EWL_ICON_FOLDER_VISITING "folder-visiting"
• #define EWL_ICON_FONT_X_GENERIC "font-x-generic"
• #define EWL_ICON_FORMAT_INDENT_LESS "format-indent-less"
• #define EWL_ICON_FORMAT_INDENT_MORE "format-indent-more"
• #define EWL_ICON_FORMAT_JUSTIFY_CENTER "format-justify-center"
• #define EWL_ICON_FORMAT_JUSTIFY_FILL "format-justify-fill"
• #define EWL_ICON_FORMAT_JUSTIFY_LEFT "format-justify-left"
• #define EWL_ICON_FORMAT_JUSTIFY_RIGHT "format-justify-right"
• #define EWL_ICON_FORMAT_TEXT_BOLD "format-text-bold"
• #define EWL_ICON_FORMAT_TEXT_DIRECTION_LTR "format-text-direction-ltr"
• #define EWL_ICON_FORMAT_TEXT_DIRECTION_RTL "format-text-direction-rtl"
• #define EWL_ICON_FORMAT_TEXT_ITALIC "format-text-italic"
• #define EWL_ICON_FORMAT_TEXT_STRIKETHROUGH "format-text-strikethrough"
• #define EWL_ICON_FORMAT_TEXT_UNDERLINE "format-text-underline"
• #define EWL_ICON_GO_BOTTOM "go-bottom"
• #define EWL_ICON_GO_DOWN "go-down"
• #define EWL_ICON_GO_FIRST "go-first"
• #define EWL_ICON_GO_HOME "go-home"
• #define EWL_ICON_GO_JUMP "go-jump"
• #define EWL_ICON_GO_LAST "go-last"

```

- #define EWL_ICON_GO_NEXT "go-next"
- #define EWL_ICON_GO_PREVIOUS "go-previous"
- #define EWL_ICON_GO_TOP "go-top"
- #define EWL_ICON_GO_UP "go-up"
- #define EWL_ICON_HELP_ABOUT "help-about"
- #define EWL_ICON_HELP_BROWSER "help-browser"
- #define EWL_ICON_HELP_CONTENTS "help-contents"
- #define EWL_ICON_HELP_FAQ "help-faq"
- #define EWL_ICON_IMAGE_LOADING "image-loading"
- #define EWL_ICON_IMAGE_MISSING "image-missing"
- #define EWL_ICON_IMAGE_X_GENERIC "image-x-generic"
- #define EWL_ICON_INPUT_GAMING "input-gaming"
- #define EWL_ICON_INPUT_KEYBOARD "input-keyboard"
- #define EWL_ICON_INPUT_MOUSE "input-mouse"
- #define EWL_ICON_INSERT_IMAGE "insert-image"
- #define EWL_ICON_INSERT_LINK "insert-link"
- #define EWL_ICON_INSERT_OBJECT "insert-object"
- #define EWL_ICON_INSERT_TEXT "insert-text"
- #define EWL_ICON_LIST_ADD "list-add"
- #define EWL_ICON_LIST_REMOVE "list-remove"
- #define EWL_ICON_MAIL_ATTACHMENT "mail-attachment"
- #define EWL_ICON_MAIL_FORWARD "mail-forward"
- #define EWL_ICON_MAIL_MARK_IMPORTANT "mail-mark-important"
- #define EWL_ICON_MAIL_MARK_JUNK "mail-mark-junk"
- #define EWL_ICON_MAIL_MARK_NOTJUNK "mail-mark-notjunk"
- #define EWL_ICON_MAIL_MARK_READ "mail-mark-read"
- #define EWL_ICON_MAIL_MARK_UNREAD "mail-mark-unread"
- #define EWL_ICON_MAIL_MESSAGE_NEW "mail-message-new"
- #define EWL_ICON_MAIL_READ "mail-read"
- #define EWL_ICON_MAIL_REPLIED "mail-replied"
- #define EWL_ICON_MAIL_REPLY_ALL "mail-reply-all"
- #define EWL_ICON_MAIL_REPLY_SENDER "mail-reply-sender"
- #define EWL_ICON_MAIL_SEND_RECEIVE "mail-send-receive"
- #define EWL_ICON_MAIL_SIGNED "mail-signed"
- #define EWL_ICON_MAIL_SIGNED_VERIFIED "mail-signed-verified"
- #define EWL_ICON_MAIL_UNREAD "mail-unread"
- #define EWL_ICON_MEDIA_CDROM "media-cdrom"
- #define EWL_ICON_MEDIA_EJECT "media-eject"
- #define EWL_ICON_MEDIA_FLOPPY "media-floppy"
- #define EWL_ICON_MEDIA_PLAYBACK_PAUSE "media-playback-pause"
- #define EWL_ICON_MEDIA_PLAYBACK_START "media-playback-start"
- #define EWL_ICON_MEDIA_PLAYBACK_STOP "media-playback-stop"
- #define EWL_ICON_MEDIA_PLAYLIST_REPEAT "media-playlist-repeat"
- #define EWL_ICON_MEDIA_PLAYLIST_SHUFFLE "media-playlist-shuffle"
- #define EWL_ICON_MEDIA_RECORD "media-record"
- #define EWL_ICON_MEDIA_SEEK_BACKWARD "media-seek-backward"
- #define EWL_ICON_MEDIA_SEEK_FORWARD "media-seek-forward"
- #define EWL_ICON_MEDIA_SKIP_BACKWARD "media-skip-backward"
- #define EWL_ICON_MEDIA_SKIP_FORWARD "media-skip-forward"
- #define EWL_ICON_MULTIMEDIA_PLAYER "multimedia-player"

```

• #define EWL_ICON_MULTIMEDIA_VOLUME_CONTROL "multimedia-
  volume-control"
• #define EWL_ICON_NETWORK_ERROR "network-error"
• #define EWL_ICON_NETWORK_IDLE "network-idle"
• #define EWL_ICON_NETWORK_OFFLINE "network-offline"
• #define EWL_ICON_NETWORK_RECEIVE "network-receive"
• #define EWL_ICON_NETWORK_SERVER "network-server"
• #define EWL_ICON_NETWORK_TRANSMIT "network-transmit"
• #define EWL_ICON_NETWORK_TRANSMIT_RECEIVE "network-transmit-
  receive"
• #define EWL_ICON_NETWORK_WIRED "network-wired"
• #define EWL_ICON_NETWORK_WIRELESS "network-wireless"
• #define EWL_ICON_NETWORK_WORKGROUP "network-workgroup"
• #define EWL_ICON_PACKAGE_X_GENERIC "package-x-generic"
• #define EWL_ICON_PREFERENCES_DESKTOP "preferences-desktop"
• #define EWL_ICON_PREFERENCES_DESKTOP_ACCESSIBILITY "preferences-desktop-accessibility"
• #define EWL_ICON_PREFERENCES_DESKTOP_ACCESSIBILITY "preferences-desktop-accessibility"
• #define EWL_ICON_PREFERENCES_DESKTOP_FONT "preferences-desktop-
  font"
• #define EWL_ICON_PREFERENCES_DESKTOP_KEYBOARD "preferences-
  desktop-keyboard"
• #define EWL_ICON_PREFERENCES_DESKTOP_LOCALE "preferences-
  desktop-locale"
• #define EWL_ICON_PREFERENCES_DESKTOP_MULTIMEDIA "preferences-desktop-multimedia"
• #define EWL_ICON_PREFERENCES_DESKTOP_PERIPHERALS "preferences-desktop-peripherals"
• #define EWL_ICON_PREFERENCES_DESKTOP_PERSONAL "preferences-
  desktop-personal"
• #define EWL_ICON_PREFERENCES_DESKTOP_SCREENSAVER "preferences-desktop-screensaver"
• #define EWL_ICON_PREFERENCES_DESKTOP_THEME "preferences-
  desktop-theme"
• #define EWL_ICON_PREFERENCES_DESKTOP_WALLPAPER "preferences-desktop-wallpaper"
• #define EWL_ICON_PREFERENCES_OTHER "preferences-other"
• #define EWL_ICON_PREFERENCES_SYSTEM "preferences-system"
• #define EWL_ICON_PREFERENCES_SYSTEM_NETWORK "preferences-
  system-network"
• #define EWL_ICON_PRINTER "printer"
• #define EWL_ICON_PRINTER_ERROR "printer-error"
• #define EWL_ICON_PRINTER_PRINTING "printer-printing"
• #define EWL_ICON_PROCESS_WORKING "process-working"
• #define EWL_ICON_SIZE_LARGE 24
• #define EWL_ICON_SIZE_MEDIUM 22
• #define EWL_ICON_SIZE_SMALL 16
• #define EWL_ICON_SOFTWARE_UPDATE_AVAILABLE "software-update-
  available"

```

- `#define EWL_ICON_SOFTWARE_UPDATE_URGENT` "software-update-urgent"
- `#define EWL_ICON_START_HERE` "start-here"
- `#define EWL_ICON_SYNC_ERROR` "sync-error"
- `#define EWL_ICON_SYNC_SYNCHRONIZING` "sync-synchronizing"
- `#define EWL_ICON_SYSTEM_FILE_MANAGER` "system-file-manager"
- `#define EWL_ICON_SYSTEM_HELP` "system-help"
- `#define EWL_ICON_SYSTEM_LOCK_SCREEN` "system-lock-screen"
- `#define EWL_ICON_SYSTEM_LOG_OUT` "system-log-out"
- `#define EWL_ICON_SYSTEM_RUN` "system-run"
- `#define EWL_ICON_SYSTEM_SEARCH` "system-search"
- `#define EWL_ICON_SYSTEM_SOFTWARE_UPDATE` "system-software-update"
- `#define EWL_ICON_TASK_DUE` "task-due"
- `#define EWL_ICON_TASK_PASSED_DUE` "task-passed-due"
- `#define EWL_ICON_TEXT_HTML` "text-html"
- `#define EWL_ICON_TEXT_X_GENERIC` "text-x-generic"
- `#define EWL_ICON_TEXT_X_GENERIC_TEMPLATE` "text-x-generic-template"
- `#define EWL_ICON_TEXT_X_SCRIPT` "text-x-script"
- `#define EWL_ICON_TOOLS_CHECK_SPELLING` "tools-check-spelling"
- `#define EWL_ICON_USER_AWAY` "user-away"
- `#define EWL_ICON_USER_DESKTOP` "user-desktop"
- `#define EWL_ICON_USER_HOME` "user-home"
- `#define EWL_ICON_USER_IDLE` "user-idle"
- `#define EWL_ICON_USER_OFFLINE` "user-offline"
- `#define EWL_ICON_USER_ONLINE` "user-online"
- `#define EWL_ICON_USER_TRASH` "user-trash"
- `#define EWL_ICON_USER_TRASH_FULL` "user-trash-full"
- `#define EWL_ICON_UTILITIES_TERMINAL` "utilities-terminal"
- `#define EWL_ICON_VIDEO_X_GENERIC` "video-x-generic"
- `#define EWL_ICON_VIEW_FULLSCREEN` "view-fullscreen"
- `#define EWL_ICON_VIEW_REFRESH` "view-refresh"
- `#define EWL_ICON_VIEW_SORT_ASCENDING` "view-sort-ascending"
- `#define EWL_ICON_VIEW_SORT_DESCENDING` "view-sort-descending"
- `#define EWL_ICON_WEATHER_CLEAR` "weather-clear"
- `#define EWL_ICON_WEATHER_CLEAR_NIGHT` "weather-clear-night"
- `#define EWL_ICON_WEATHER_FEW_CLOUDS` "weather-few-clouds"
- `#define EWL_ICON_WEATHER_FEW_CLOUDS_NIGHT` "weather-few-clouds-night"
- `#define EWL_ICON_WEATHER_FOG` "weather-fog"
- `#define EWL_ICON_WEATHER_OVERCAST` "weather-overcast"
- `#define EWL_ICON_WEATHER_SEVERE_ALERT` "weather-severe-alert"
- `#define EWL_ICON_WEATHER_SHOWERS` "weather-showers"
- `#define EWL_ICON_WEATHER_SHOWERS_SCATTERED` "weather-showers-scattered"
- `#define EWL_ICON_WEATHER_SNOW` "weather-snow"
- `#define EWL_ICON_WEATHER_STORM` "weather-storm"
- `#define EWL_ICON_WINDOW_CLOSE` "window-close"
- `#define EWL_ICON_WINDOW_NEW` "window-new"

- `#define EWL_ICON_X_OFFICE_ADDRESS_BOOK "x-office-address-book"`
- `#define EWL_ICON_X_OFFICE_CALENDAR "x-office-calendar"`
- `#define EWL_ICON_X_OFFICE_DOCUMENT "x-office-document"`
- `#define EWL_ICON_X_OFFICE_PRESENTATION "x-office-presentation"`
- `#define EWL_ICON_X_OFFICE_SPREADSHEET "x-office-spreadsheet"`
- `#define EWL_ICON_ZOOM_BEST_FIT "zoom-best-fit"`
- `#define EWL_ICON_ZOOM_IN "zoom-in"`
- `#define EWL_ICON_ZOOM_ORIGINAL "zoom-original"`
- `#define EWL_ICON_ZOOM_OUT "zoom-out"`

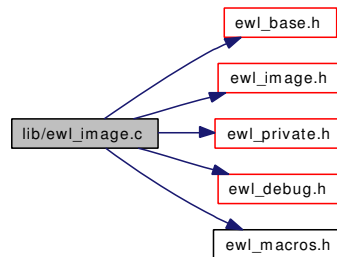
Functions

- `const char * ewl_icon_theme_icon_path_get (const char *icon, int size)`
Retrives the full path to the specified icon, or NULL if none found.
- `int ewl_icon_theme_init (void)`
Initializes the icon theme system.
- `void ewl_icon_theme_shutdown (void)`
Shuts down the icon theme system.
- `void ewl_icon_theme_theme_change (void)`
Called when the icon theme is changed so we can clean up any caching we have in place.

8.115 lib/ewl_image.c File Reference

```
#include "ewl_base.h"
#include "ewl_image.h"
#include "ewl_private.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
```

Include dependency graph for ewl_image.c:



Functions

- void **ewl_image_cb_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_image_cb_destroy** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_image_cb_mouse_down** (Ewl_Widget *w, void *ev_data, void *user_data __UNUSED__)
- void **ewl_image_cb_mouse_move** (Ewl_Widget *w, void *ev_data, void *user_data __UNUSED__)
- void **ewl_image_cb_mouse_up** (Ewl_Widget *w, void *ev_data, void *user_data __UNUSED__)
- void **ewl_image_cb_obscure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_image_cb_reparent** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_image_cb_reveal** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- unsigned int **ewl_image_constrain_get** (Ewl_Image *i)

Get the current constrain value set on the image.

- void **ewl_image_constrain_set** (Ewl_Image *i, unsigned int size)

Set a size which, if the image is bigger than, scale proportionally.

- const char * **ewl_image_file_key_get** (Ewl_Image *i)
- void **ewl_image_file_key_set** (Ewl_Image *i, const char *key)

Sets the key to use for the image.

- const char * **ewl_image_file_path_get** (Ewl_Image *i)

get the filename this image uses

- void **ewl_image_file_path_set** (**Ewl_Image** *i, const char *path)
Sets the path to the image file.
- void **ewl_image_file_set** (**Ewl_Image** *i, const char *im, const char *key)
Change the image file displayed by an image widget.
- void **ewl_image_flip** (**Ewl_Image** *img, **Ewl_Orientation** orient)
Flips the given image in the given direction.
- int **ewl_image_init** (**Ewl_Image** *i)
Initialize an image widget to default values and callbacks.
- **Ewl_Widget** * **ewl_image_new** (void)
Load an image widget with specified image contents.
- char **ewl_image_proportional_get** (**Ewl_Image** *i)
Get boolean to determine how to scale.
- void **ewl_image_proportional_set** (**Ewl_Image** *i, char p)
Set boolean to determine how to scale.
- void **ewl_image_rotate** (**Ewl_Image** *i, **Ewl_Rotate** rotate)
Rotates the given image by the given rotate value.
- void **ewl_image_scale_get** (**Ewl_Image** *i, double *wp, double *hp)
Retrieve the percentage an image is scaled.
- void **ewl_image_scale_set** (**Ewl_Image** *i, double wp, double hp)
Scale image dimensions by a percentage.
- void **ewl_image_size_get** (**Ewl_Image** *i, int *w, int *h)
Scale image dimensions to a specific size.
- void **ewl_image_size_set** (**Ewl_Image** *i, int w, int h)
Scale image dimensions to a specific size.
- **Ewl_Widget** * **ewl_image_thumbnail_get** (**Ewl_Image** *i)
Create a widget representing a thumbnailed version of the image.
- int **ewl_image_thumbnail_init** (**Ewl_Image_Thumbnail** *image)
*Initialize an **Ewl_Image_Thumbnail** (p. 689) to default values.*
- **Ewl_Widget** * **ewl_image_thumbnail_new** (void)
*This will create and initialize a new **Ewl_Image_Thumbnail** (p. 689) widget.*
- void **ewl_image_thumbnail_request** (**Ewl_Image_Thumbnail** *thumb, const char *path)
This will request the generation of a thumbnail for the image given by path.

- void ewl_image_tile_set (Ewl_Image *i, int x, int y, int w, int h)
Tile the image with the given start position and given size.

- Ewl_View * ewl_image_view_get (void)
*Creates and returns a view to be used by **Ewl_Image** (p. 686) widgets.*

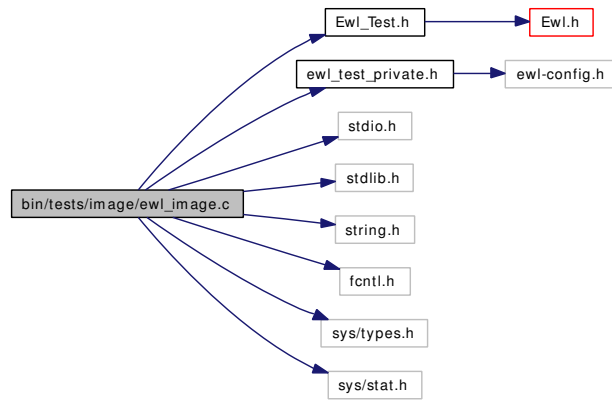
8.115.1 Function Documentation

- 8.115.1.1 void ewl_image_cb_configure (Ewl_Widget * w, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.115.1.2 void ewl_image_cb_destroy (Ewl_Widget * w, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.115.1.3 void ewl_image_cb_mouse_down (Ewl_Widget * w, void * ev_data,
void *user_data __ *UNUSED* __)
- 8.115.1.4 void ewl_image_cb_mouse_move (Ewl_Widget * w, void * ev_data,
void *user_data __ *UNUSED* __)
- 8.115.1.5 void ewl_image_cb_mouse_up (Ewl_Widget * w, void * ev_data, void
*user_data __ *UNUSED* __)
- 8.115.1.6 void ewl_image_cb_obscure (Ewl_Widget * w, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.115.1.7 void ewl_image_cb_reparent (Ewl_Widget * w, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.115.1.8 void ewl_image_cb_reveal (Ewl_Widget * w, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)

8.116 bin/tests/image/ewl_image.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
```

Include dependency graph for ewl_image.c:



Functions

- `void test_info (Ewl_Test *test)`

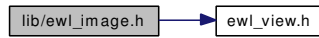
8.116.1 Function Documentation

8.116.1.1 `void test_info (Ewl_Test * test)`

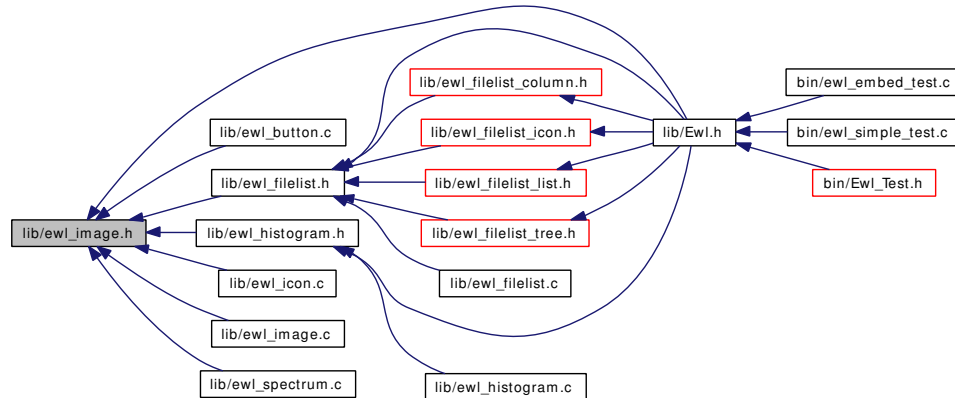
8.117 lib/ewl_image.h File Reference

```
#include "ewl_view.h"
```

Include dependency graph for ewl_image.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Image**
*Inherits from **Ewl_Widget** (p. 785) and extends to provide an image widget.*
- struct **Ewl_Image_Thumbnail**
*Inherits from **Ewl_Image** (p. 686) and extends to provide reference to original image.*

Defines

- #define **EWL_IMAGE**(image) ((**Ewl_Image** *) image)
- #define **EWL_IMAGE_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_IMAGE_TYPE))
- #define **EWL_IMAGE_THUMBNAIL**(image) ((**Ewl_Image_Thumbnail** *) image)
- #define **EWL_IMAGE_THUMBNAIL_TYPE** "image_thumbnail"
- #define **EWL_IMAGE_TYPE** "image"

Typedefs

- typedef **Ewl_Image** **Ewl_Image**
- typedef **Ewl_Image_Thumbnail** **Ewl_Image_Thumbnail**

Functions

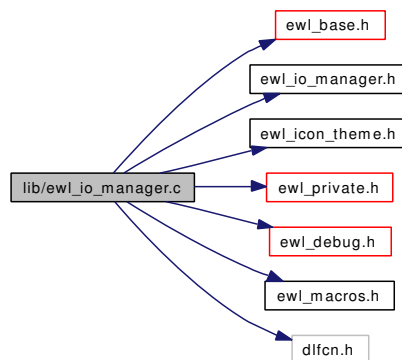
- void **ewl_image_cb_configure** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_image_cb_destroy** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_image_cb_mouse_down** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_image_cb_mouse_move** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_image_cb_mouse_up** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_image_cb_obscure** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_image_cb_reveal** (Ewl_Widget *w, void *ev_data, void *user_data)
- unsigned int **ewl_image_constrain_get** (Ewl_Image *i)
Get the current constrain value set on the image.
- void **ewl_image_constrain_set** (Ewl_Image *i, unsigned int size)
Set a size which, if the image is bigger than, scale proportionally.
- const char * **ewl_image_file_key_get** (Ewl_Image *i)
- void **ewl_image_file_key_set** (Ewl_Image *i, const char *key)
Sets the key to use for the image.
- const char * **ewl_image_file_path_get** (Ewl_Image *i)
get the filename this image uses
- void **ewl_image_file_path_set** (Ewl_Image *i, const char *im)
Sets the path to the image file.
- void **ewl_image_file_set** (Ewl_Image *i, const char *im, const char *key)
Change the image file displayed by an image widget.
- void **ewl_image_flip** (Ewl_Image *img, Ewl_Orientation orient)
Flips the given image in the given direction.
- int **ewl_image_init** (Ewl_Image *i)
Initialize an image widget to default values and callbacks.
- Ewl_Widget * **ewl_image_new** (void)
Load an image widget with specified image contents.
- char **ewl_image_proportional_get** (Ewl_Image *i)
Get boolean to determine how to scale.
- void **ewl_image_proportional_set** (Ewl_Image *i, char p)
Set boolean to determine how to scale.
- void **ewl_image_rotate** (Ewl_Image *i, Ewl_Rotate rotate)
Rotates the given image by the given rotate value.
- void **ewl_image_scale_get** (Ewl_Image *i, double *wp, double *hp)
Retrieve the percentage an image is scaled.

- void **ewl_image_scale_set** (**Ewl_Image** *i, double wp, double hp)
Scale image dimensions by a percentage.
- void **ewl_image_size_get** (**Ewl_Image** *i, int *w, int *h)
Scale image dimensions to a specific size.
- void **ewl_image_size_set** (**Ewl_Image** *i, int w, int h)
Scale image dimensions to a specific size.
- **Ewl_Widget** * **ewl_image_thumbnail_get** (**Ewl_Image** *i)
Create a widget representing a thumbnail version of the image.
- int **ewl_image_thumbnail_init** (**Ewl_Image_Thumbnail** *image)
*Initialize an **Ewl_Image_Thumbnail** (p. 689) to default values.*
- **Ewl_Widget** * **ewl_image_thumbnail_new** (void)
*This will create and initialize a new **Ewl_Image_Thumbnail** (p. 689) widget.*
- void **ewl_image_thumbnail_request** (**Ewl_Image_Thumbnail** *thumb, const char *path)
This will request the generation of a thumbnail for the image given by path.
- void **ewl_image_tile_set** (**Ewl_Image** *i, int x, int y, int w, int h)
Tile the image with the given start position and given size.
- **Ewl_View** * **ewl_image_view_get** (void)
*Creates and returns a view to be used by **Ewl_Image** (p. 686) widgets.*

8.118 lib/ewl_io_manager.c File Reference

```
#include "ewl_base.h"
#include "ewl_io_manager.h"
#include "ewl_icon_theme.h"
#include "ewl_private.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include <dlfcn.h>
```

Include dependency graph for ewl_io_manager.c:



Functions

- `const char * ewl_io_manager_extension_icon_name_get (const char *ext)`
Retrieve the icon name for the given extension or NULL if none found.
- `int ewl_io_manager_init (void)`
Initializes the IO manager system.
- `const char * ewl_io_manager_mime_type_icon_name_get (const char *mime)`
Retrives the icon name for the given mime type or NULL if none found.
- `void ewl_io_manager_shutdown (void)`
Shuts down the io manager system.
- `Ewl_Widget * ewl_io_manager_string_read (const char *string, const char *mime)`
Reads the given string and interprets it as the given mime type.
- `int ewl_io_manager_string_write (Ewl_Widget *data, char **string, const char *mime)`
Writes the given data into the given string pointer.
- `const char * ewl_io_manager_uri_mime_type_get (const char *uri)`

Retrives the mime type for the given URI or NULL on error.

- **Ewl_Widget * ewl_io_manager_uri_read** (const char *uri)
Creates a widget to display the URI contents. Returns NULL on error.
- **int ewl_io_manager_uri_write** (Ewl_Widget *data, const char *uri, const char *mime)
Writes the given data out to uri as mime type mime.

8.118.1 Function Documentation

8.118.1.1 **const char* ewl_io_manager_extension_icon_name_get** (const char **ext*)

Retrieve the icon name for the given extension or NULL if none found.

Parameters:

ext,: The extension to look up

Returns:

Returns the icon name for the given extension or NULL if none found

8.118.1.2 **int ewl_io_manager_init** (void)

Initializes the IO manager system.

Returns:

Returns TRUE on successful initialization, FALSE otherwise

8.118.1.3 **const char* ewl_io_manager_mime_type_icon_name_get** (const char **mime*)

Retrives the icon name for the given mime type or NULL if none found.

Parameters:

mime,: The mime type to get the icon for

Returns:

Returns the icon name for the given mime type or NULL if none found

8.118.1.4 **void ewl_io_manager_shutdown** (void)

Shuts down the io manager system.

Returns:

Returns no value

8.118.1.5 Ewl_Widget* ewl_io_manager_string_read (const char * *string*, const char * *mime*)

Reads the given string and interprets it as the given mime type.

Parameters:

string,: The string to read

mime,: The mime type to interpret the string as

Returns:

Returns a widget representing the string in the given mime type

8.118.1.6 int ewl_io_manager_string_write (Ewl_Widget * *data*, char ** *string*, const char * *mime*)

Writes the given data into the given string pointer.

Parameters:

data,: The data to write

string,: Where to write

mime,: The mime type to write as

Returns:

Returns TRUE if the write is successful, FALSE otherwise

8.118.1.7 const char* ewl_io_manager_uri_mime_type_get (const char * *uri*)

Retrives the mime type for the given URI or NULL on error.

Parameters:

uri,: The URI to look up the mime type for

Returns:

Returns the mime type for the given URI or NULL on error

8.118.1.8 Ewl_Widget* ewl_io_manager_uri_read (const char * *uri*)

Creates a widget to display the URI contents. Returns NULL on error.

Parameters:

uri,: The URI to read

Returns:

Returns a widget displaying the URI contents or NULL on error

8.118.1.9 `int ewl_io_manager_uri_write (Ewl_Widget * data, const char * uri,
const char * mime)`

Writes the given *data* out to *uri* as mime type *mime*.

Parameters:

data,: The data to write

uri,: The URI to write the data too

mime,: The mime type to write the data out as

Returns:

Returns TRUE on success, FALSE otherwise

8.119 bin/tests/io_manager/ewl_io_manager.c File Reference

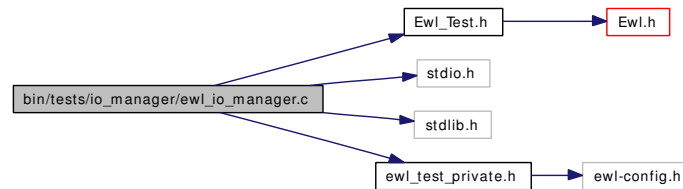
```
#include "Ewl_Test.h"
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include "ewl_test_private.h"
```

Include dependency graph for ewl_io_manager.c:



Functions

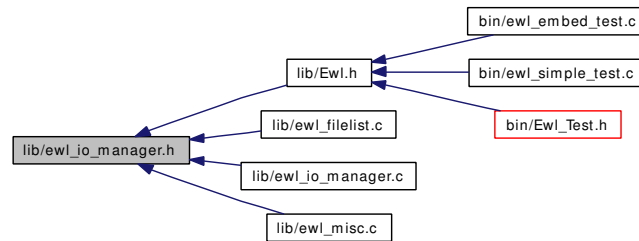
- `void test_info (Ewl_Test *test)`

8.119.1 Function Documentation

8.119.1.1 `void test_info (Ewl_Test * test)`

8.120 lib/ewl_io_manager.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_IO_Manager_Plugin**
Provides the base structure for IO Manager plugins.

Typedefs

- typedef **Ewl_IO_Manager_Plugin** **Ewl_IO_Manager_Plugin**

Functions

- const char * **ewl_io_manager_extension_icon_name_get** (const char *ext)
Retrieve the icon name for the given extension or NULL if none found.
- int **ewl_io_manager_init** (void)
Initializes the IO manager system.
- const char * **ewl_io_manager_mime_type_icon_name_get** (const char *mime)
Retrives the icon name for the given mime type or NULL if none found.
- void **ewl_io_manager_shutdown** (void)
Shuts down the io manager system.
- **Ewl_Widget** * **ewl_io_manager_string_read** (const char *string, const char *mime)
Reads the given string and interprets it as the given mime type.
- int **ewl_io_manager_string_write** (**Ewl_Widget** *data, char **string, const char *mime)
Writes the given data into the given string pointer.
- const char * **ewl_io_manager_uri_mime_type_get** (const char *uri)
Retrives the mime type for the given URI or NULL on error.
- **Ewl_Widget** * **ewl_io_manager_uri_read** (const char *uri)

Creates a widget to display the URI contents. Returns NULL on error.

- **int ewl_io_manager_uri_write** (**Ewl_Widget** *data, const char *uri, const char *mime)

Writes the given data out to uri as mime type mime.

8.120.1 Typedef Documentation

8.120.1.1 **typedef struct Ewl_IO_Manager_Plugin Ewl_IO_Manager_Plugin**

Ewl_IO_Manager_Plugin (p. 690)

8.120.2 Function Documentation

8.120.2.1 **const char* ewl_io_manager_extension_icon_name_get** (const char **ext*)

Retrieve the icon name for the given extension or NULL if none found.

Parameters:

ext,: The extension to look up

Returns:

Returns the icon name for the given extension or NULL if none found

8.120.2.2 **int ewl_io_manager_init** (void)

Initializes the IO manager system.

Returns:

Returns TRUE on successful initialization, FALSE otherwise

8.120.2.3 **const char* ewl_io_manager_mime_type_icon_name_get** (const char **mime*)

Retrives the icon name for the given mime type or NULL if none found.

Parameters:

mime,: The mime type to get the icon for

Returns:

Returns the icon name for the given mime type or NULL if none found

8.120.2.4 void ewl_io_manager_shutdown (void)

Shuts down the io manager system.

Returns:

Returns no value

8.120.2.5 Ewl_Widget* ewl_io_manager_string_read (const char * *string*, const char * *mime*)

Reads the given string and interprets it as the given mime type.

Parameters:

string,: The string to read

mime,: The mime type to interpret the string as

Returns:

Returns a widget representing the string in the given mime type

8.120.2.6 int ewl_io_manager_string_write (Ewl_Widget * *data*, char ** *string*, const char * *mime*)

Writes the given data into the given string pointer.

Parameters:

data,: The data to write

string,: Where to write

mime,: The mime type to write as

Returns:

Returns TRUE if the write is successful, FALSE otherwise

8.120.2.7 const char* ewl_io_manager_uri_mime_type_get (const char * *uri*)

Retrives the mime type for the given URI or NULL on error.

Parameters:

uri,: The URI to look up the mime type for

Returns:

Returns the mime type for the given URI or NULL on error

8.120.2.8 Ewl_Widget* ewl_io_manager_uri_read (const char * *uri*)

Creates a widget to display the URI contents. Returns NULL on error.

Parameters:

uri,: The URI to read

Returns:

Returns a widget displaying the URI contents or NULL on error

8.120.2.9 int ewl_io_manager_uri_write (Ewl_Widget * *data*, const char * *uri*, const char * *mime*)

Writes the given *data* out to *uri* as mime type *mime*.

Parameters:

data,: The data to write

uri,: The URI to write the data too

mime,: The mime type to write the data out as

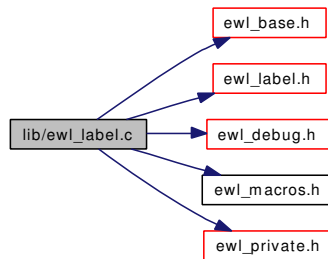
Returns:

Returns TRUE on success, FALSE otherwise

8.121 lib/ewl_label.c File Reference

```
#include "ewl_base.h"
#include "ewl_label.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_label.c:



Functions

- **int ewl_label_init (Ewl_Label *la)**
Initializes the la widget.
- **Ewl_Widget * ewl_label_new (void)**
*Creates a new **Ewl_Label** (p. 692) widget with the text text in it.*
- **const char * ewl_label_text_get (Ewl_Label *la)**
*Gets the current text set into the label. Note, this pointer will only be valid until the next time **ewl_label_text_set()** (p. 289) is called.*
- **void ewl_label_text_set (Ewl_Label *la, const char *text)**
Sets the given text into the widget la.
- **Ewl_View * ewl_label_view_get (void)**
*Creates and returns a view to be used by **Ewl_Label** (p. 692) widgets.*

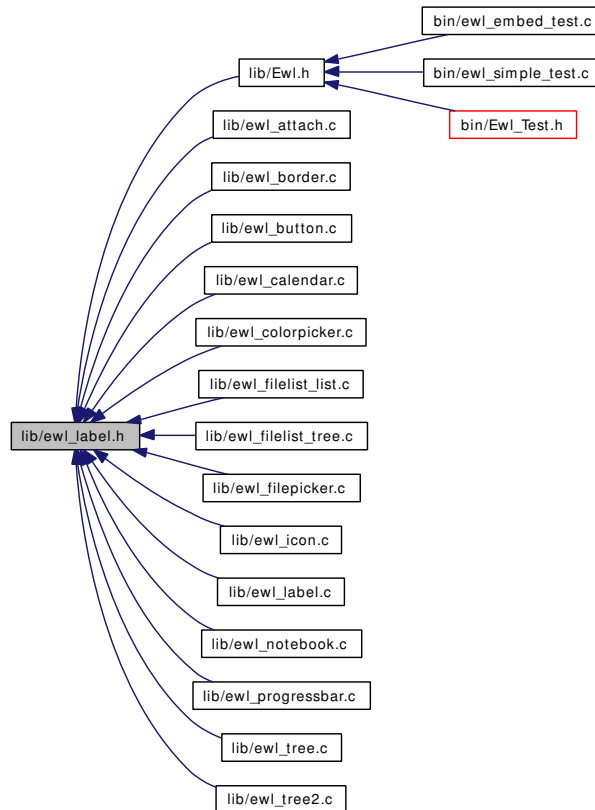
8.122 lib/ewl_label.h File Reference

```
#include "ewl_view.h"
```

Include dependency graph for ewl_label.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Label**

*Inherits from **Ewl_Widget** (p. 785) class to provide a simple label.*

Defines

- `#define EWL_LABEL(label) ((Ewl_Label *)label)`
- `#define EWL_LABEL_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_LABEL_TYPE))`
- `#define EWL_LABEL_TYPE "label"`

Typedefs

- typedef **Ewl_Label** **Ewl_Label**

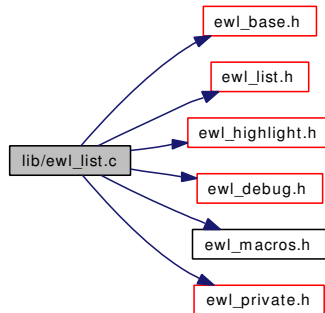
Functions

- int **ewl_label_init** (**Ewl_Label** *la)
Initializes the la widget.
- **Ewl_Widget** * **ewl_label_new** (void)
*Creates a new **Ewl_Label** (p. 692) widget with the text text in it.*
- const char * **ewl_label_text_get** (**Ewl_Label** *la)
*Gets the current text set into the label. Note, this pointer will only be valid until the next time **ewl_label_text_set**() (p. 289) is called.*
- void **ewl_label_text_set** (**Ewl_Label** *la, const char *text)
Sets the given text into the widget la.
- **Ewl_View** * **ewl_label_view_get** (void)
*Creates and returns a view to be used by **Ewl_Label** (p. 692) widgets.*

8.123 lib/ewl_list.c File Reference

```
#include "ewl_base.h"
#include "ewl_list.h"
#include "ewl_highlight.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_list.c:



Functions

- void ewl_list_cb_child_add (Ewl_Container *c, Ewl_Widget *w)
- void ewl_list_cb_configure (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void ewl_list_cb_item_clicked (Ewl_Widget *w, void *ev __UNUSED__, void *data)
- void ewl_list_cb_selected_change (Ewl_MVC *mvc)
- int ewl_list_init (Ewl_List *list)

*Initialises an **Ewl_List** (p. 693) widget to default values.*
- Ewl_Widget * ewl_list_new (void)

*Creates and initializes a new **Ewl_List** (p. 693) widget.*

8.123.1 Function Documentation

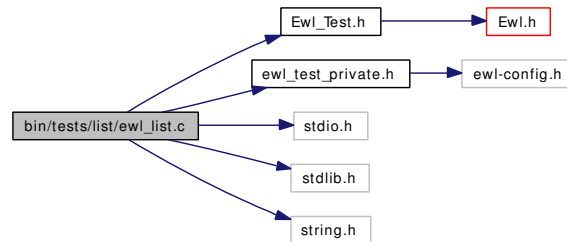
8.123.1.1 void ewl_list_cb_configure (Ewl_Widget * w, void *ev __UNUSED__, void *data __UNUSED__)

8.123.1.2 void ewl_list_cb_item_clicked (Ewl_Widget * w, void *ev __UNUSED__, void * data)

8.124 bin/tests/list/ewl_list.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for ewl_list.c:



Data Structures

- struct `List_Test_Data`
- struct `List_Test_Row_Data`

Typedefs

- typedef `List_Test_Data` `List_Test_Data`
- typedef `List_Test_Row_Data` `List_Test_Row_Data`

Functions

- void `test_info` (`Ewl_Test *test`)

8.124.1 Typedef Documentation

8.124.1.1 typedef struct `List_Test_Data` `List_Test_Data`

8.124.1.2 typedef struct `List_Test_Row_Data` `List_Test_Row_Data`

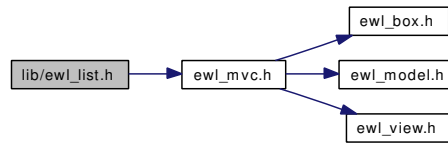
8.124.2 Function Documentation

8.124.2.1 void `test_info` (`Ewl_Test * test`)

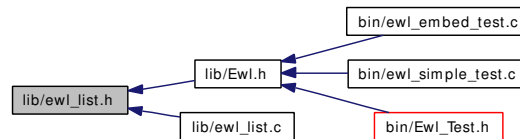
8.125 lib/ewl_list.h File Reference

```
#include "ewl_mvc.h"
```

Include dependency graph for ewl_list.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_List**

*Inherits from **Ewl_MVC** (p. 705) and extends to provide a list widget.*

Defines

- `#define EWL_LIST(list) ((Ewl_List *)list)`
- `#define EWL_LIST_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_LIST_TYPE))`
- `#define EWL_LIST_TYPE "list"`

Typedefs

- `typedef Ewl_List Ewl_List`

Functions

- `void ewl_list_cb_child_add (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_list_cb_configure (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_list_cb_item_clicked (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_list_cb_selected_change (Ewl_MVC *mvc)`
- `int ewl_list_init (Ewl_List *list)`

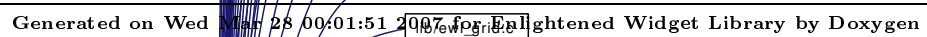
*Initialises an **Ewl_List** (p. 693) widget to default values.*

- `Ewl_Widget * ewl_list_new (void)`

*Creates and initializes a new **Ewl_List** (p. 693) widget.*

8.126 lib/ewl_macros.h File Reference

This graph shows which files directly or indirectly include this file:



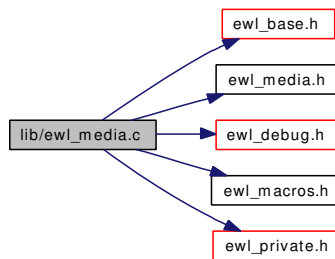
Defines

- `#define FREE(dat)`
- `#define IF_FREE(dat)`
- `#define IF_FREE_HASH(hash)`
- `#define IF_FREE_LIST(list)`
- `#define IF_RELEASE(item)`
- `#define MAX(x, y) (((x) > (y)) ? (x) : (y))`
- `#define MIN(x, y) (((x) > (y)) ? (y) : (x))`
- `#define NEW(type, num) calloc(num, sizeof(type));`
- `#define REALLOC(dat, type, num)`
- `#define ZERO(ptr, type, num) ptr = memset(ptr, 0, sizeof(type) * (num))`

8.127 lib/ewl_media.c File Reference

```
#include "ewl_base.h"
#include "ewl_media.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_media.c:



Functions

- **int ewl_media_audio_mute_get** (Ewl_Media *m)
Checks if the media widget is muted.
- **void ewl_media_audio_mute_set** (Ewl_Media *m, int mute)
Mutes the media widget.
- **double ewl_media_audio_volume_get** (Ewl_Media *m)
Gets the current volume from the media widget.
- **void ewl_media_audio_volume_set** (Ewl_Media *m, double v)
Sets the media widget to the given volume.
- **void ewl_media_cb_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- **void ewl_media_cb_realize** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- **void ewl_media_cb_reveal** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- **void ewl_media_cb_unrealize** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- **int ewl_media_init** (Ewl_Media *m)
Initialize the fields and callbacks of a media object.
- **unsigned int ewl_media_is_available** (void)
This will check if the modules required for the media widget were available at compile time or not.

- `int ewl_media_length_get (Ewl_Media *m)`
Retrieve the length of the media displayed by the media widget.
- `void ewl_media_length_time_get (Ewl_Media *m, int *h, int *min, double *s)`
Puts the length of the video into the h, m, s variables.
- `const char * ewl_media_media_get (Ewl_Media *m)`
Retrieve the media of a media widget.
- `void ewl_media_media_set (Ewl_Media *m, const char *media)`
Set the media of a media widget.
- `Ewl_Media_Module_Type ewl_media_module_get (Ewl_Media *m)`
Retrieve the module of a media widget.
- `int ewl_media_module_set (Ewl_Media *m, Ewl_Media_Module_Type module)`
Set the module of a media widget.
- `Ewl_Widget * ewl_media_new (void)`
Allocate a new media widget.
- `void ewl_media_play_set (Ewl_Media *m, int p)`
Sets the media widget into the given state.
- `double ewl_media_position_get (Ewl_Media *m)`
Returns the position of the current media.
- `void ewl_media_position_set (Ewl_Media *m, double p)`
Sets the media widget to the specified position.
- `void ewl_media_position_time_get (Ewl_Media *m, int *h, int *min, double *s)`
Puts the position of the video into the h, m, s variables.
- `int ewl_media_seekable_get (Ewl_Media *m)`
Returns if the media area is seekable.

8.127.1 Function Documentation

8.127.1.1 `void ewl_media_cb_configure (Ewl_Widget * w, void *ev_data
__ UNUSED __, void *user_data __ UNUSED __)`

8.127.1.2 `void ewl_media_cb_realize (Ewl_Widget * w, void *ev_data
__ UNUSED __, void *user_data __ UNUSED __)`

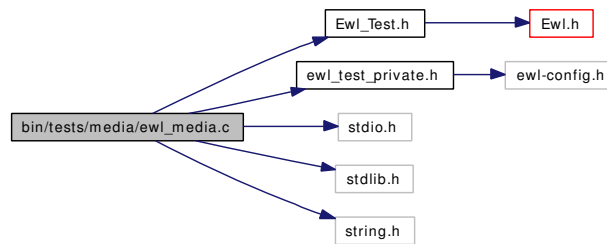
8.127.1.3 `void ewl_media_cb_reveal (Ewl_Widget * w, void *ev_data
__ UNUSED __, void *user_data __ UNUSED __)`

8.127.1.4 `void ewl_media_cb_unrealize (Ewl_Widget * w, void *ev_data
__ UNUSED __, void *user_data __ UNUSED __)`

8.128 bin/tests/media/ewl_media.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for ewl_media.c:



Data Structures

- struct **Control**

Defines

- `#define PATH_MAX 1024`

Functions

- void `test_info (Ewl_Test *test)`

8.128.1 Define Documentation

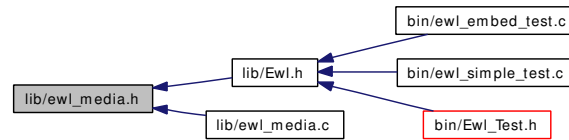
8.128.1.1 `#define PATH_MAX 1024`

8.128.2 Function Documentation

8.128.2.1 void `test_info (Ewl_Test * test)`

8.129 lib/ewl_media.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Media**

*Inherits from the **Ewl_Widget** (p. 785) class and extends it to provide for multi-line media layout, obstacle wrapping, and a variety of formatting.*

Defines

- `#define EWL_MEDIA(media) ((Ewl_Media *) media)`
- `#define EWL_MEDIA_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_MEDIA_TYPE))`
- `#define EWL_MEDIA_TYPE "media"`

Typedefs

- `typedef Ewl_Media Ewl_Media`

Functions

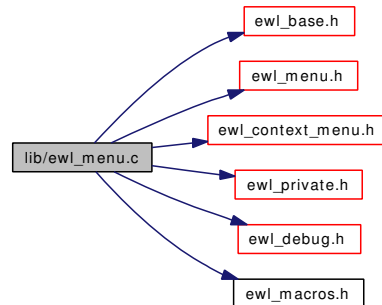
- `int ewl_media_audio_mute_get (Ewl_Media *m)`
Checks if the media widget is muted.
- `void ewl_media_audio_mute_set (Ewl_Media *m, int mute)`
Mutes the media widget.
- `double ewl_media_audio_volume_get (Ewl_Media *m)`
Gets the current volume from the media widget.
- `void ewl_media_audio_volume_set (Ewl_Media *m, double v)`
Sets the media widget to the given volume.
- `void ewl_media_cb_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_media_cb_realize (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_media_cb_reveal (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_media_cb_unrealize (Ewl_Widget *w, void *ev_data, void *user_data)`
- `int ewl_media_init (Ewl_Media *m)`
Initialize the fields and callbacks of a media object.

- unsigned int **ewl_media_is_available** (void)
This will check if the modules required for the media widget were available at compile time or not.
- int **ewl_media_length_get** (Ewl_Media *m)
Retrieve the length of the media displayed by the media widget.
- void **ewl_media_length_time_get** (Ewl_Media *m, int *h, int *min, double *s)
Puts the length of the video into the h, m, s variables.
- const char * **ewl_media_media_get** (Ewl_Media *m)
Retrieve the media of a media widget.
- void **ewl_media_media_set** (Ewl_Media *m, const char *media)
Set the media of a media widget.
- Ewl_Media_Module_Type **ewl_media_module_get** (Ewl_Media *m)
Retrieve the module of a media widget.
- int **ewl_media_module_set** (Ewl_Media *m, Ewl_Media_Module_Type module)
Set the module of a media widget.
- Ewl_Widget * **ewl_media_new** (void)
Allocate a new media widget.
- void **ewl_media_play_set** (Ewl_Media *m, int p)
Sets the media widget into the given state.
- double **ewl_media_position_get** (Ewl_Media *m)
Returns the position of the current media.
- void **ewl_media_position_set** (Ewl_Media *m, double p)
Sets the media widget to the specified position.
- void **ewl_media_position_time_get** (Ewl_Media *m, int *h, int *min, double *s)
Puts the position of the video into the h, m, s variables.
- int **ewl_media_seekable_get** (Ewl_Media *m)
Returns if the media area is seekable.

8.130 lib/ewl_menu.c File Reference

```
#include "ewl_base.h"
#include "ewl_menu.h"
#include "ewl_context_menu.h"
#include "ewl_private.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
```

Include dependency graph for ewl_menu.c:



Functions

- void **ewl_menu_cb_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_menu_cb_destroy** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_menu_cb_expand** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_menu_cb_mouse_move** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_menu_cb_popup_destroy** (Ewl_Widget *w __UNUSED__, void *ev __UNUSED__, void *data)
- void **ewl_menu_collapse** (Ewl_Menu *menu)

Collapses the popup portion of the menu.

- void **ewl_menu_from_info** (Ewl_Menu *menu, Ewl_Menu_Info *info)

Initializes menu with info.

- int **ewl_menu_init** (Ewl_Menu *menu)

Initialize an internal menu to starting values.

- int **ewl_menu_mouse_feed** (Ewl_Menu *menu, int x, int y)
- Ewl_Widget * **ewl_menu_new** (void)

Create a new internal menu.

8.130.1 Function Documentation

- 8.130.1.1 `void ewl_menu_cb_configure (Ewl_Widget * w, void *ev_data
__ UNUSED __, void *user_data __ UNUSED __)`
- 8.130.1.2 `void ewl_menu_cb_destroy (Ewl_Widget * w, void *ev
__ UNUSED __, void *data __ UNUSED __)`
- 8.130.1.3 `void ewl_menu_cb_expand (Ewl_Widget * w, void *ev_data
__ UNUSED __, void *user_data __ UNUSED __)`
- 8.130.1.4 `void ewl_menu_cb_mouse_move (Ewl_Widget * w, void *ev_data
__ UNUSED __, void *user_data __ UNUSED __)`
- 8.130.1.5 `void ewl_menu_cb_popup_destroy (Ewl_Widget *w __ UNUSED __,
void *ev __ UNUSED __, void * data)`

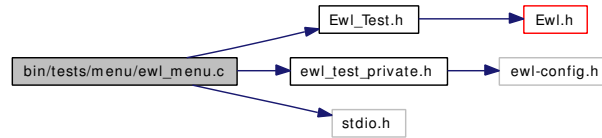
8.131 bin/tests/menu/ewl_menu.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_menu.c:



Functions

- `void test_info (Ewl_Test *test)`

8.131.1 Function Documentation

8.131.1.1 `void test_info (Ewl_Test * test)`

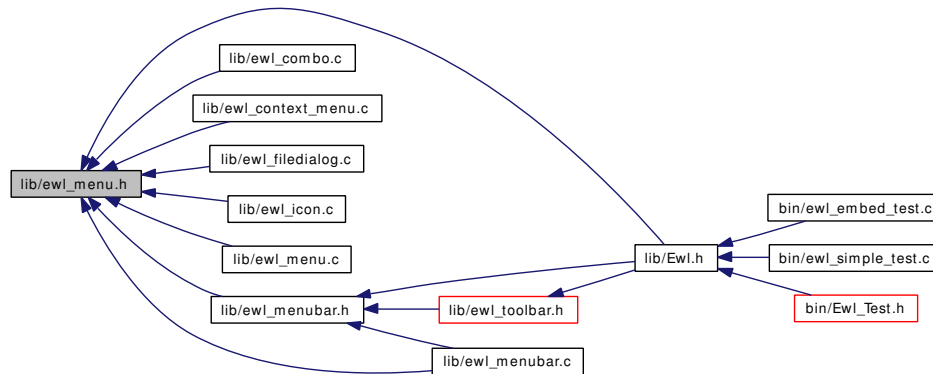
8.132 lib/ewl_menu.h File Reference

```
#include "ewl_menu_item.h"
```

Include dependency graph for ewl_menu.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Menu**

Inherits from the Ewl_Menu_Base and extends to provide a simple menu.

- struct **Ewl_Menu_Info**

Provides a simple structure to contain menu information.

Defines

- `#define EWL_MENU(menu) ((Ewl_Menu *) menu)`
- `#define EWL_MENU_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_MENU_TYPE))`
- `#define EWL_MENU_TYPE "menu"`

Typedefs

- `typedef Ewl_Menu Ewl_Menu`
- `typedef Ewl_Menu_Info Ewl_Menu_Info`

Functions

- void **ewl_menu_cb_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_menu_cb_destroy** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_menu_cb_expand** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_menu_cb_hide** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_menu_cb_mouse_move** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_menu_cb_popup_destroy** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_menu_cb_realize** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_menu_collapse** (**Ewl_Menu** *menu)

Collapses the popup portion of the menu.

- void **ewl_menu_from_info** (**Ewl_Menu** *menu, **Ewl_Menu_Info** *info)

Initializes menu with info.

- int **ewl_menu_init** (**Ewl_Menu** *menu)

Initialize an internal menu to starting values.

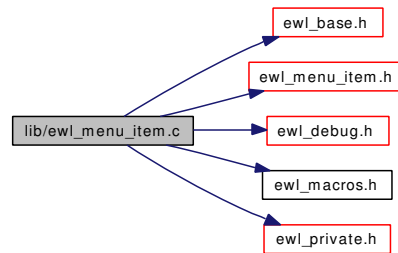
- int **ewl_menu_mouse_feed** (**Ewl_Menu** *menu, int x, int y)
- **Ewl_Widget** * **ewl_menu_new** (void)

Create a new internal menu.

8.133 lib/ewl_menu_item.c File Reference

```
#include "ewl_base.h"
#include "ewl_menu_item.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_menu_item.c:



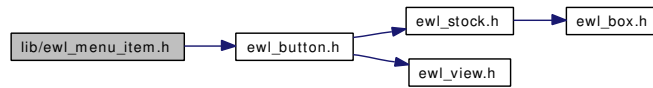
Functions

- **int ewl_menu_item_init (Ewl_Menu_Item *item)**
Initialize the fields of a menu item to their defaults.
- **Ewl_Widget * ewl_menu_item_new (void)**
Create a new menu item to place in a menu.

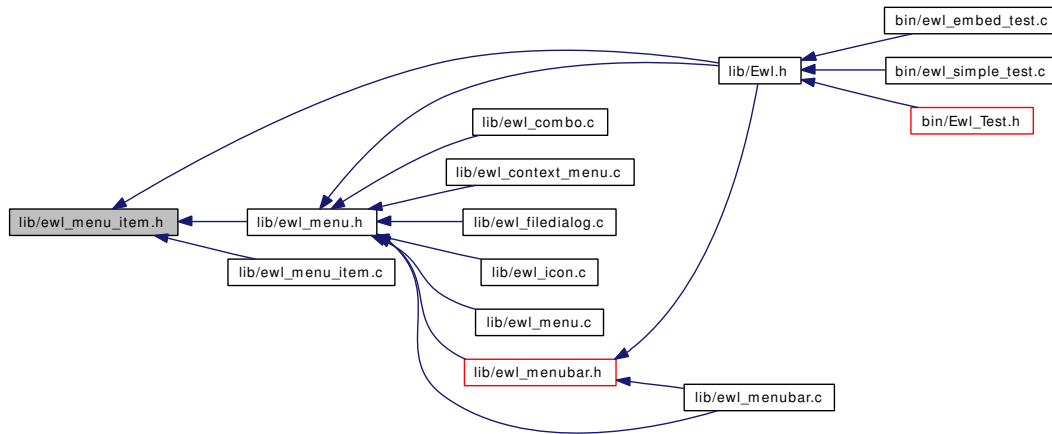
8.134 lib/ewl_menu_item.h File Reference

```
#include "ewl_button.h"
```

Include dependency graph for ewl_menu_item.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Menu_Item**

*Inherits from **Ewl_Button** (p. 589) and provides a basic menu item widget.*

Defines

- #define **EWL_MENU_ITEM(mi)** ((**Ewl_Menu_Item** *) mi)
- #define **EWL_MENU_ITEM_IS(w)** (ewl_widget_type_is(EWL_WIDGET(w), EWL_MENU_ITEM_TYPE))
- #define **EWL_MENU_ITEM_TYPE** "menu_item"

Typedefs

- typedef **Ewl_Menu_Item** **Ewl_Menu_Item**

Functions

- int **ewl_menu_item_init** (**Ewl_Menu_Item** *menu)

Initialize the fields of a menu item to their defaults.

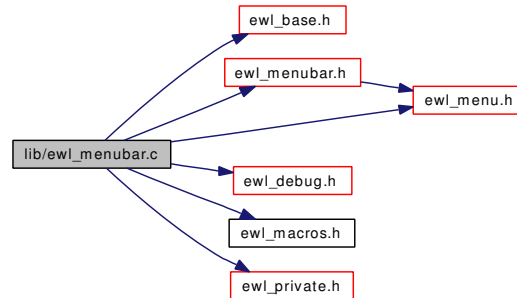
- `Ewl_Widget * ewl_menu_item_new (void)`

Create a new menu item to place in a menu.

8.135 lib/ewl_menubar.c File Reference

```
#include "ewl_base.h"
#include "ewl_menubar.h"
#include "ewl_menu.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_menubar.c:



Functions

- **Ewl_Widget * ewl_hmenubar_new** (void)
*Creates a new **Ewl_Menubar** (p. 701) in a horizontal orientation.*
- void **ewl_menubar_cb_child_add** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_menubar_from_info** (Ewl_Menubar *mb, Ewl_Menubar_Info *info)
Sets up the menubar mb with the given info.
- int **ewl_menubar_init** (Ewl_Menubar *mb)
Initialize a menubar to default values.
- **Ewl_Widget * ewl_menubar_new** (void)
Allocate and initialize a new menubar widget.
- **Ewl_Orientation ewl_menubar_orientation_get** (Ewl_Menubar *mb)
Gets the orientation of the menubar.
- void **ewl_menubar_orientation_set** (Ewl_Menubar *mb, Ewl_Orientation o)
Sets the orientation of the menubar.
- **Ewl_Widget * ewl_vmenubar_new** (void)
*Creates a new **Ewl_Menubar** (p. 701) in a vertical orientation.*

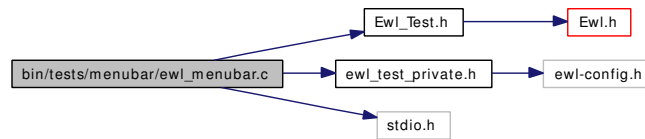
8.136 bin/tests/menubar/ewl_menubar.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_menubar.c:



Functions

- `void test_info (Ewl_Test *test)`

8.136.1 Function Documentation

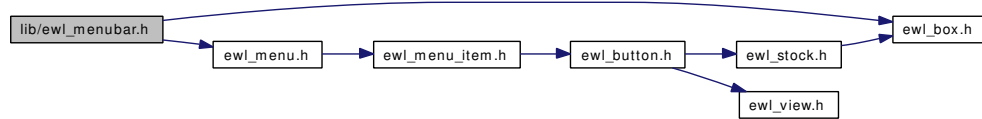
8.136.1.1 `void test_info (Ewl_Test * test)`

8.137 lib/ewl_menubar.h File Reference

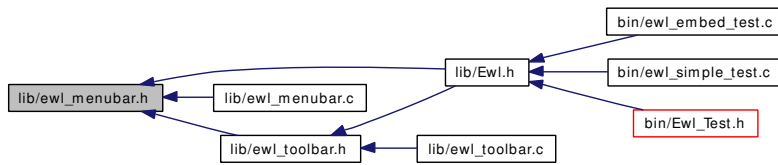
```
#include "ewl_box.h"
```

```
#include "ewl_menu.h"
```

Include dependency graph for ewl_menubar.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Menubar**

*Inherits from the **Ewl_Box** (p. 586) class and extends it to provide for a menubar.*

- struct **Ewl_Menubar_Info**

Provides a simple menu structure for the menubar.

Defines

- #define **EWL_MENUBAR**(menubar) ((**Ewl_Menubar** *) menubar)
- #define **EWL_MENUBAR_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_MENUBAR_TYPE))
- #define **EWL_MENUBAR_TYPE** "menubar"

Typedefs

- typedef **Ewl_Menubar** **Ewl_Menubar**
- typedef **Ewl_Menubar_Info** **Ewl_Menubar_Info**

Functions

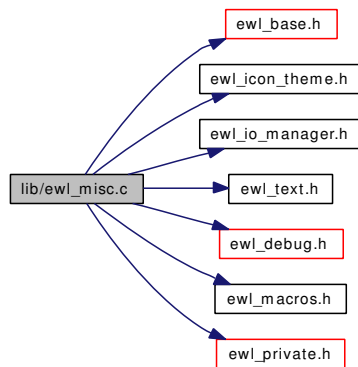
- **Ewl_Widget** * **ewl_hmenubar_new** (void)
*Creates a new **Ewl_Menubar** (p. 701) in a horizontal orientation.*
- void **ewl_menubar_cb_child_add** (**Ewl_Container** *c, **Ewl_Widget** *w)

- `void ewl__menubar__from__info (Ewl__Menubar *mb, Ewl__Menubar__Info *info)`
Sets up the menubar mb with the given info.
- `int ewl__menubar__init (Ewl__Menubar *mb)`
Initialize a menubar to default values.
- `Ewl__Widget * ewl__menubar__new (void)`
Allocate and initialize a new menubar widget.
- `Ewl__Orientation ewl__menubar__orientation__get (Ewl__Menubar *mb)`
Gets the orientation of the menubar.
- `void ewl__menubar__orientation__set (Ewl__Menubar *mb, Ewl__Orientation o)`
Sets the orientation of the menubar.
- `Ewl__Widget * ewl__vmenubar__new (void)`
*Creates a new **Ewl__Menubar** (p. 701) in a vertical orientation.*

8.138 lib/ewl_misc.c File Reference

```
#include "ewl_base.h"
#include "ewl_icon_theme.h"
#include "ewl_io_manager.h"
#include "ewl_text.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_misc.c:



Data Structures

- struct **Ewl_Configure_Queue**

Defines

- `#define EWL_CONFIGURE_QUEUE_SIZE 4092`
- `#define EWL_GC_LIMIT 300`

Typedefs

- `typedef Ewl_Configure_Queue Ewl_Configure_Queue`

Functions

- void **ewl_backtrace** (void)
This will print a backtrace at the given point.
- void **ewl_canvas_destroy** (void *evas)
- void **ewl_canvas_object_destroy** (void *obj)
- void **ewl_configure_request** (Ewl_Widget *w)
Ask for a widget to be configured during idle loop.

- void **ewl_debug_indent_print** (int mod_dir)

Creates a string used to indent debug messages.

- void **ewl_destroy_request** (Ewl_Widget *w)

- int **ewl_init** (int *argc, char **argv)

Initialize the internal variables of ewl to begin the program.

- void **ewl_main** (void)

The main execution loop of EWL.

- void **ewl_main_quit** (void)

Notifies ewl to quit at the end of this pass of the main loop.

- void **ewl_print_help** (void)

Print out the Ewl help text.

- void **ewl_print_warning** (void)

This is used by debugging macros for breakpoints.

- void **ewl_realize_cancel_request** (Ewl_Widget *w)

- void **ewl_realize_request** (Ewl_Widget *w)

- void **ewl_segv** (void)

This will cause EWL to SEGV. (Handy for debugging).

- int **ewl_shutdown** (void)

Cleanup internal data structures used by ewl.

- void **ewl_shutdown_add** (Ewl_Shutdown_Hook hook)

Allows a widget to hookup a shutdown callback that will be called when Ewl is shutdown.

Variables

- Ecore_List * **configure_active** = NULL
- Ecore_List * **configure_available** = NULL
- Ecore_List * **ewl_embed_list**
- Ecore_List * **ewl_window_list**

8.138.1 Define Documentation

8.138.1.1 `#define EWL_CONFIGURE_QUEUE_SIZE 4092`

8.138.1.2 `#define EWL_GC_LIMIT 300`

8.138.2 Typedef Documentation

8.138.2.1 `typedef struct Ewl_Configure_Queue Ewl_Configure_Queue`

8.138.3 Function Documentation

8.138.3.1 `void ewl_backtrace (void)`

This will print a backtrace at the given point.

Returns:

Returns no value.

8.138.3.2 `void ewl_print_warning (void)`

This is used by debugging macros for breakpoints.

Returns:

Returns no value.

Set a breakpoint at this function in order to retrieve backtraces from warning messages.

8.138.3.3 `void ewl_segv (void)`

This will cause EWL to SEGV. (Handy for debugging).

Returns:

Returns no value.

8.138.4 Variable Documentation

8.138.4.1 `Ecore_List* configure_active = NULL`

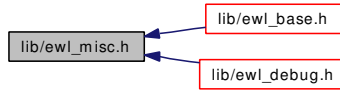
8.138.4.2 `Ecore_List* configure_available = NULL`

8.138.4.3 `Ecore_List* ewl_embed_list`

8.138.4.4 `Ecore_List* ewl_window_list`

8.139 lib/ewl_misc.h File Reference

This graph shows which files directly or indirectly include this file:



Typedefs

- typedef void(*) **Ewl_Shutdown_Hook** (void)

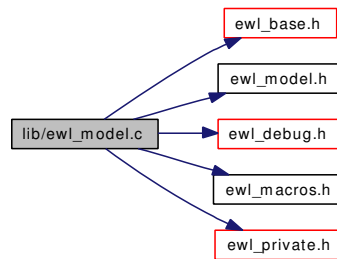
Functions

- void **ewl_canvas_destroy** (void *evas)
- void **ewl_canvas_object_destroy** (void *obj)
- void **ewl_configure_request** (Ewl_Widget *w)
Ask for a widget to be configured during idle loop.
- void **ewl_debug_indent_print** (int mod_dir)
Creates a string used to indent debug messages.
- void **ewl_destroy_request** (Ewl_Widget *w)
- int **ewl_init** (int *argc, char **argv)
Initialize the internal variables of ewl to begin the program.
- void **ewl_main** (void)
The main execution loop of EWL.
- void **ewl_main_quit** (void)
Notifies ewl to quit at the end of this pass of the main loop.
- void **ewl_print_help** (void)
Print out the Ewl help text.
- void **ewl_realize_cancel_request** (Ewl_Widget *w)
- void **ewl_realize_request** (Ewl_Widget *w)
- int **ewl_shutdown** (void)
Cleanup internal data structures used by ewl.
- void **ewl_shutdown_add** (Ewl_Shutdown_Hook hook)
Allows a widget to hookup a shutdown callback that will be called when Ewl is shutdown.

8.140 lib/ewl_model.c File Reference

```
#include "ewl_base.h"
#include "ewl_model.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_model.c:



Functions

- **int ewl_model_cb_ecore_list_count** (void *data)
This will return the number of elements in the ecore_list.
- **void * ewl_model_cb_ecore_list_fetch** (void *data, unsigned int row, unsigned int col __UNUSED__)
This will return the row data element from the list.
- **Ewl_Model_Column_Sortable ewl_model_column_sortable_get** (Ewl_Model *m)
Retrieves the sortable callback on the model.
- **void ewl_model_column_sortable_set** (Ewl_Model *m, Ewl_Model_Column_Sortable sortable)
Sets the sortable callback.
- **Ewl_Model_Data_Count ewl_model_data_count_get** (Ewl_Model *m)
Gets the count callback from the model.
- **void ewl_model_data_count_set** (Ewl_Model *m, Ewl_Model_Data_Count count)
Sets the count callback into the model.
- **Ewl_Model_Data_Expandable ewl_model_data_expandable_get** (Ewl_Model *m)
Retrieves the expandable callback set on the model.
- **void ewl_model_data_expandable_set** (Ewl_Model *m, Ewl_Model_Data_Expandable exp)

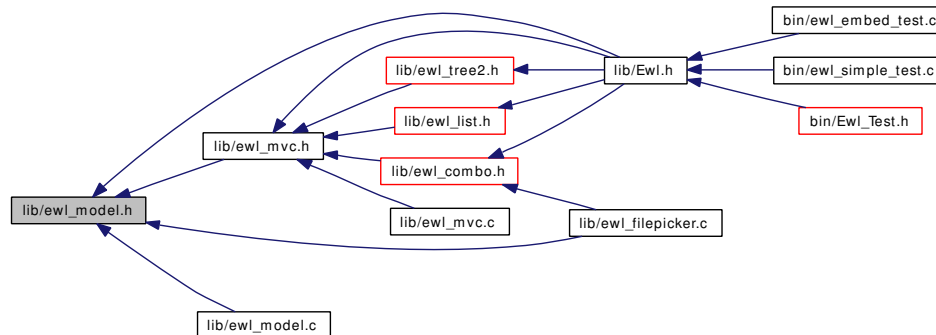
Sets the expandable callback on the given model.

- **Ewl_Model_Data_Fetch** `ewl_model_data_fetch_get (Ewl_Model *m)`
Gets the fetch callback from the model.
- `void ewl_model_data_fetch_set (Ewl_Model *m, Ewl_Model_Data_Fetch get)`
Sets the fetch callback in the model.
- **Ewl_Model_Data_Free** `ewl_model_data_free_get (Ewl_Model *m)`
Get the data free pointer set on this model.
- `void ewl_model_data_free_set (Ewl_Model *m, Ewl_Model_Data_Free data_free)`
Set the data free pointer on this view.
- **Ewl_Model_Data_Header_Fetch** `ewl_model_data_header_fetch_get (Ewl_Model *m)`
Retrieves the header fetch function.
- `void ewl_model_data_header_fetch_set (Ewl_Model *m, Ewl_Model_Data_Header_Fetch get)`
Sets the fetch callback into the model.
- **Ewl_Model_Data_Sort** `ewl_model_data_sort_get (Ewl_Model *m)`
Gets the sort callback from the model.
- `void ewl_model_data_sort_set (Ewl_Model *m, Ewl_Model_Data_Sort sort)`
Sets the sort callback into the model.
- **Ewl_Model *ewl_model_ecore_list_get (void)**
Retrieves a model pre-initialized to work with an ecore list. This will setup the fetch and count methods for you.
- **Ewl_Model_Expansion_Data_Fetch** `ewl_model_expansion_data_fetch_get (Ewl_Model *m)`
Gets the expansion_data callback from the model.
- `void ewl_model_expansion_data_fetch_set (Ewl_Model *m, Ewl_Model_Expansion_Data_Fetch get)`
Sets the subfetch callback into the model.
- **Ewl_Model_Expansion_Model_Fetch** `ewl_model_expansion_model_fetch_get (Ewl_Model *m)`
Retrieves the model expansion model fetch function.
- `void ewl_model_expansion_model_fetch_set (Ewl_Model *m, Ewl_Model_Expansion_Model_Fetch f)`
Sets the model expansion model fetch callback to f.

- `int ewl_model_init (Ewl_Model *model)`
Set the model to the default values.
- `Ewl_Model * ewl_model_new (void)`
*Creates a new **Ewl_Model** (p. 703) object.*

8.141 lib/ewl_model.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Model**
Holds the callbacks needed to define a model.

Defines

- `#define EWL_MODEL(model) ((Ewl_Model *)model)`
- `#define EWL_MODEL_COLUMN_SORTABLE(f) ((Ewl_Model_Column_Sortable)f)`
- `#define EWL_MODEL_DATA_COUNT(f) ((Ewl_Model_Data_Count)f)`
- `#define EWL_MODEL_DATA_EXPANDABLE(f) ((Ewl_Model_Data_Expandable)f)`
- `#define EWL_MODEL_DATA_FETCH(f) ((Ewl_Model_Data_Fetch)f)`
- `#define EWL_MODEL_DATA_HEADER_FETCH(f) ((Ewl_Model_Data_Header_Fetch)f)`
- `#define EWL_MODEL_DATA_SORT(f) ((Ewl_Model_Data_Sort)f)`
- `#define EWL_MODEL_EXPANSION_DATA_FETCH(f) ((Ewl_Model_Expansion_Data_Fetch)f)`
- `#define EWL_MODEL_EXPANSION_MODEL_FETCH(f) ((Ewl_Model_Expansion_Model_Fetch)f)`
- `#define EWL_MODEL_FREE(f) ((Ewl_Model_Data_Free)f)`

Typedefs

- `typedef Ewl_Model Ewl_Model`
- `typedef int(*) Ewl_Model_Column_Sortable (void *data, int col)`
- `typedef int(*) Ewl_Model_Data_Count (void *data)`
- `typedef int(*) Ewl_Model_Data_Expandable (void *data, unsigned int row)`
- `typedef void *(*) Ewl_Model_Data_Fetch (void *data, unsigned int row, unsigned int column)`

- typedef void(*) **Ewl_Model_Data_Free** (void *data)
- typedef void *(*) **Ewl_Model_Data_Header_Fetch** (void *data, unsigned int col)
- typedef void(*) **Ewl_Model_Data_Sort** (void *data, unsigned int column, **Ewl_Sort_Direction** sort)
- typedef void *(*) **Ewl_Model_Expansion_Data_Fetch** (void *data, unsigned int row)
- typedef **Ewl_Model** *(*) **Ewl_Model_Expansion_Model_Fetch** (void *data, unsigned int row)

Functions

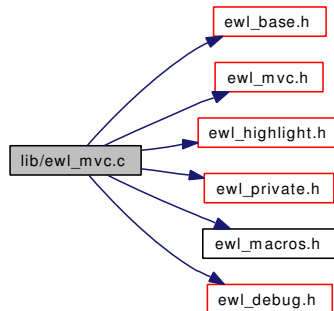
- int **ewl_model_cb_ecore_list_count** (void *data)
This will return the number of elements in the ecore_list.
- void * **ewl_model_cb_ecore_list_fetch** (void *data, unsigned int row, unsigned int col)
This will return the row data element from the list.
- **Ewl_Model_Column_Sortable** **ewl_model_column_sortable_get** (**Ewl_Model** *m)
Retrieves the sortable callback on the model.
- void **ewl_model_column_sortable_set** (**Ewl_Model** *m, **Ewl_Model_Column_Sortable** sortable)
Sets the sortable callback.
- **Ewl_Model_Data_Count** **ewl_model_data_count_get** (**Ewl_Model** *m)
Gets the count callback from the model.
- void **ewl_model_data_count_set** (**Ewl_Model** *m, **Ewl_Model_Data_Count** count)
Sets the count callback into the model.
- **Ewl_Model_Data_Expandable** **ewl_model_data_expandable_get** (**Ewl_Model** *m)
Retrieves the expandable callback set on the model.
- void **ewl_model_data_expandable_set** (**Ewl_Model** *m, **Ewl_Model_Data_Expandable** exp)
Sets the expandable callback on the given model.
- **Ewl_Model_Data_Fetch** **ewl_model_data_fetch_get** (**Ewl_Model** *m)
Gets the fetch callback from the model.
- void **ewl_model_data_fetch_set** (**Ewl_Model** *m, **Ewl_Model_Data_Fetch** get)
Sets the fetch callback in the model.
- **Ewl_Model_Data_Free** **ewl_model_data_free_get** (**Ewl_Model** *m)
Get the data free pointer set on this model.

- `void ewl_model_data_free_set (Ewl_Model *m, Ewl_Model_Data_Free assign)`
Set the data free pointer on this view.
- `Ewl_Model_Data_Header_Fetch ewl_model_data_header_fetch_get (Ewl_Model *m)`
Retrieves the header fetch function.
- `void ewl_model_data_header_fetch_set (Ewl_Model *m, Ewl_Model_Data_Header_Fetch get)`
Sets the fetch callback into the model.
- `Ewl_Model_Data_Sort ewl_model_data_sort_get (Ewl_Model *m)`
Gets the sort callback from the model.
- `void ewl_model_data_sort_set (Ewl_Model *m, Ewl_Model_Data_Sort sort)`
Sets the sort callback into the model.
- `Ewl_Model * ewl_model_ecore_list_get (void)`
Retrieves a model pre-initialized to work with an ecore list. This will setup the fetch and count methods for you.
- `Ewl_Model_Expansion_Data_Fetch ewl_model_expansion_data_fetch_get (Ewl_Model *m)`
Gets the expansion_data callback from the model.
- `void ewl_model_expansion_data_fetch_set (Ewl_Model *m, Ewl_Model_Expansion_Data_Fetch get)`
Sets the subfetch callback into the model.
- `Ewl_Model_Expansion_Model_Fetch ewl_model_expansion_model_fetch_get (Ewl_Model *m)`
Retrieves the model expansion model fetch function.
- `void ewl_model_expansion_model_fetch_set (Ewl_Model *m, Ewl_Model_Expansion_Model_Fetch f)`
Sets the model expansion model fetch callback to f.
- `int ewl_model_init (Ewl_Model *model)`
Set the model to the default values.
- `Ewl_Model * ewl_model_new (void)`
*Creates a new **Ewl_Model** (p. 703) object.*

8.142 lib/ewl_mvc.c File Reference

```
#include "ewl_base.h"
#include "ewl_mvc.h"
#include "ewl_highlight.h"
#include "ewl_private.h"
#include "ewl_macros.h"
#include "ewl_debug.h"
```

Include dependency graph for ewl_mvc.c:



Functions

- void **ewl_mvc_cb_destroy** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void * **ewl_mvc_data_get** (Ewl_MVC *mvc)
Retrieves the data set into the MVC widget.
- void **ewl_mvc_data_set** (Ewl_MVC *mvc, void *data)
Sets the given data data into the MVC widget mvc.
- unsigned int **ewl_mvc_dirty_get** (Ewl_MVC *mvc)
Retrieves the dirty status of the MVC widget.
- void **ewl_mvc_dirty_set** (Ewl_MVC *mvc, unsigned int dirty)
Sets the dirty status of the MVC widget mvc to the dirty state.
- void **ewl_mvc_handle_click** (Ewl_MVC *mvc, Ewl_Model *model, void *data, int row, int column)
- void **ewl_mvc_highlight** (Ewl_MVC *mvc, Ewl_Container *c, Ewl_Widget *(*widget)(Ewl_MVC *mvc, void *data, int row, int column))
- int **ewl_mvc_init** (Ewl_MVC *mvc)
Initializes an MVC widget ot default values.
- Ewl_Model * **ewl_mvc_model_get** (Ewl_MVC *mvc)
Retrieves the model set into the MVC widget.

- **void ewl_mvc_model_set** (Ewl_MVC *mvc, Ewl_Model *model)
Sets the given model into the tree.
- **void ewl_mvc_selected_add** (Ewl_MVC *mvc, Ewl_Model *model, void *data, int row, int column)
Adds the given index to the selected list.
- **void ewl_mvc_selected_change_cb_set** (Ewl_MVC *mvc, void(*cb)(Ewl_MVC *mvc))
- **void ewl_mvc_selected_clear** (Ewl_MVC *mvc)
clears the selection list
- **int ewl_mvc_selected_count_get** (Ewl_MVC *mvc)
Retrives the number of items selected in the widget.
- **Ewl_Selection_Idx * ewl_mvc_selected_get** (Ewl_MVC *mvc)
Retrieves the last selected item. Return must be free'd.
- **unsigned int ewl_mvc_selected_is** (Ewl_MVC *mvc, void *data __UNUSED__, int row, int column)
Checks if the given index is selected or not.
- **Ecore_List * ewl_mvc_selected_list_get** (Ewl_MVC *mvc)
Retrieves the list of selected indicies. DO NOT remove or change items in this list.
- **void ewl_mvc_selected_list_set** (Ewl_MVC *mvc, Ecore_List *list)
Sets the list of items to select. This will remove any items it needs from the list.
- **void ewl_mvc_selected_range_add** (Ewl_MVC *mvc, Ewl_Model *model, void *data, int srow, int scolumn, int erow, int ecolumn)
Sets the given range, inclusive, as selected in the mvc.
- **void ewl_mvc_selected_rm** (Ewl_MVC *mvc, void *data __UNUSED__, int row, int column)
Removes the given index from the list of selected indices.
- **void ewl_mvc_selected_set** (Ewl_MVC *mvc, Ewl_Model *model, void *data, int row, int column)
Sets the given index as selected.
- **Ewl_Selection * ewl_mvc_selection_index_new** (Ewl_Model *model, void *data, int row, int column)
Creates a new index selection based on given values.
- **Ewl_Selection_Mode ewl_mvc_selection_mode_get** (Ewl_MVC *mvc)
Retrieves the selection mode of the widget.
- **void ewl_mvc_selection_mode_set** (Ewl_MVC *mvc, Ewl_Selection_Mode mode)
Sets the selection capabilities of the mvc widget.

- **Ewl_Selection * ewl_mvc_selection_range_new** (Ewl_Model *model, void *data, int srow, int scolumn, int erow, int ecolumn)
Creates a new range selection based on given values.
- **void ewl_mvc_view_change_cb_set** (Ewl_MVC *mvc, void(*cb)(Ewl_MVC *mvc))
This callback will be called whenever the ewl_mvc_view_set routine is called to notify the inheriting widget that the view has changed.
- **Ewl_View * ewl_mvc_view_get** (Ewl_MVC *mvc)
Retrives the current view set on the MVC.
- **void ewl_mvc_view_set** (Ewl_MVC *mvc, Ewl_View *view)
Sets the given view onto the MVC.

8.142.1 Function Documentation

8.142.1.1 void ewl_mvc_cb_destroy (Ewl_Widget * w, void *ev __ *UNUSED* __, void *data __ *UNUSED* __)

8.142.1.2 unsigned int ewl_mvc_selected_is (Ewl_MVC * mvc, void *data __ *UNUSED* __, int row, int column)

Checks if the given index is selected or not.

Parameters:

mvc,: The MVC to work with
data,: *UNUSED*
row,: The row to check for
column,: The column to check for

Returns:

Returns TRUE if the index is selected, FALSE otherwise

8.142.1.3 void ewl_mvc_selected_rm (Ewl_MVC * mvc, void *data __ *UNUSED* __, int row, int column)

Removes the given index from the list of selected indices.

Parameters:

mvc,: The MVC to work with
data,: The parent data containing the index selection
row,: The row to remove
column,: The column to remove

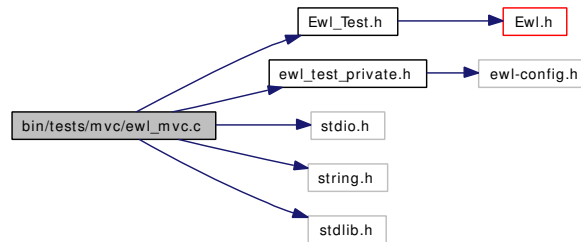
Returns:

Returns no value

8.143 bin/tests/mvc/ewl_mvc.c File Reference

```
#include "Ewl_Test.h"  
#include "ewl_test_private.h"  
#include <stdio.h>  
#include <string.h>  
#include <stdlib.h>
```

Include dependency graph for ewl_mvc.c:



Functions

- `void test_info (Ewl_Test *test)`

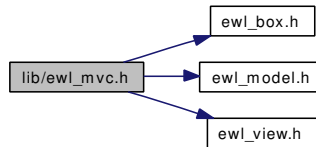
8.143.1 Function Documentation

8.143.1.1 `void test_info (Ewl_Test * test)`

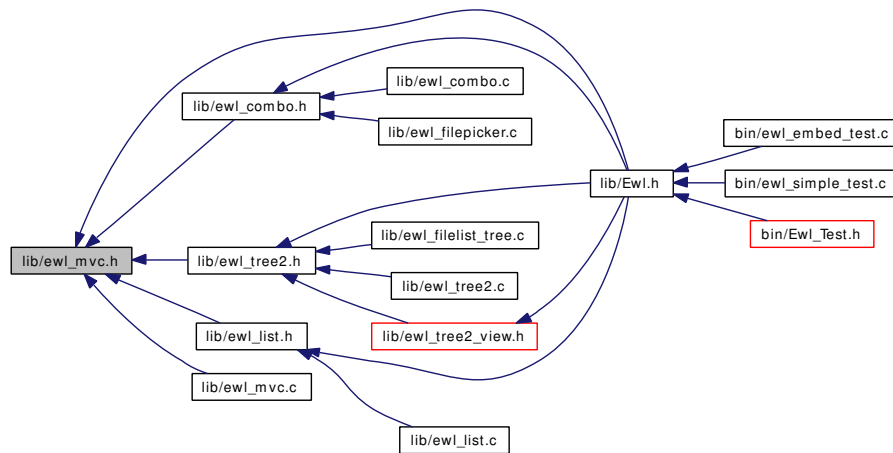
8.144 lib/ewl_mvc.h File Reference

```
#include "ewl_box.h"
#include "ewl_model.h"
#include "ewl_view.h"
```

Include dependency graph for ewl_mvc.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_MVC**

*Inherits from **Ewl_Box** (p. 586) and extends to provide mvc functionality.*

- struct **Ewl_Selection**

Structure to store information on selections.

- struct **Ewl_Selection_Idx**

Structure to store information on a single index selection.

- struct **Ewl_Selection_Range**

Structure to store a range of selected cells.

Defines

- `#define EWL_MVC(mvc) ((Ewl_MVC *)mvc)`
- `#define EWL_MVC_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_MVC_TYPE))`
- `#define EWL_MVC_TYPE "mvc"`
- `#define EWL_SELECTION(x) ((Ewl_Selection *)x)`
- `#define EWL_SELECTION_IDX(x) ((Ewl_Selection_Idx *)x)`
- `#define EWL_SELECTION_RANGE(x) ((Ewl_Selection_Range *)x)`

Typedefs

- `typedef Ewl_MVC Ewl_MVC`
- `typedef Ewl_Selection Ewl_Selection`
- `typedef Ewl_Selection_Idx Ewl_Selection_Idx`
- `typedef Ewl_Selection_Range Ewl_Selection_Range`

Functions

- `void ewl_mvc_cb_destroy (Ewl_Widget *w, void *ev, void *data)`
Retrieves the data set into the MVC widget.
- `void * ewl_mvc_data_get (Ewl_MVC *mvc)`
Sets the given data data into the MVC widget mvc.
- `unsigned int ewl_mvc_dirty_get (Ewl_MVC *mvc)`
Retrieves the dirty status of the MVC widget.
- `void ewl_mvc_dirty_set (Ewl_MVC *mvc, unsigned int dirty)`
Sets the dirty status of the MVC widget mvc to the dirty state.
- `void ewl_mvc_handle_click (Ewl_MVC *mvc, Ewl_Model *model, void *data, int row, int column)`
- `void ewl_mvc_highlight (Ewl_MVC *mvc, Ewl_Container *c, Ewl_Widget *(*widget)(Ewl_MVC *mvc, void *data, int row, int column))`
- `int ewl_mvc_init (Ewl_MVC *mvc)`
Initializes an MVC widget ot default values.
- `Ewl_Model * ewl_mvc_model_get (Ewl_MVC *mvc)`
Retrieves the model set into the MVC widget.
- `void ewl_mvc_model_set (Ewl_MVC *mvc, Ewl_Model *model)`
Sets the given model into the tree.
- `void ewl_mvc_selected_add (Ewl_MVC *mvc, Ewl_Model *model, void *data, int row, int column)`
Adds the given index to the selected list.

- void **ewl_mvc_selected_change_cb_set** (Ewl_MVC *mvc, void(*cb)(Ewl_MVC *mvc))
- void **ewl_mvc_selected_clear** (Ewl_MVC *mvc)
clears the selection list
- int **ewl_mvc_selected_count_get** (Ewl_MVC *mvc)
Retrives the number of items selected in the widget.
- Ewl_Selection_Idx * **ewl_mvc_selected_get** (Ewl_MVC *mvc)
Retrieves the last selected item. Return must be free'd.
- unsigned int **ewl_mvc_selected_is** (Ewl_MVC *mvc, void *data, int row, int column)
- Ecore_List * **ewl_mvc_selected_list_get** (Ewl_MVC *mvc)
Retrieves the list of selected indicies. DO NOT remove or change items in this list.
- void **ewl_mvc_selected_list_set** (Ewl_MVC *mvc, Ecore_List *list)
Sets the list of items to select. This will remove any items it needs from the list.
- void **ewl_mvc_selected_range_add** (Ewl_MVC *mvc, Ewl_Model *model, void *data, int srow, int scolumn, int erow, int ecolumn)
Sets the given range, inclusive, as selected in the mvc.
- void **ewl_mvc_selected_rm** (Ewl_MVC *mvc, void *data, int row, int column)
- void **ewl_mvc_selected_set** (Ewl_MVC *mvc, Ewl_Model *model, void *data, int row, int column)
Sets the given index as selected.
- Ewl_Selection * **ewl_mvc_selection_index_new** (Ewl_Model *model, void *data, int row, int column)
Creates a new index selection based on given values.
- Ewl_Selection_Mode **ewl_mvc_selection_mode_get** (Ewl_MVC *mvc)
Retrieves the selection mode of the widget.
- void **ewl_mvc_selection_mode_set** (Ewl_MVC *mvc, Ewl_Selection_Mode mode)
Sets the selection capabilities of the mvc widget.
- Ewl_Selection * **ewl_mvc_selection_range_new** (Ewl_Model *model, void *data, int srow, int scolumn, int erow, int ecolumn)
Creates a new range selection based on given values.
- void **ewl_mvc_view_change_cb_set** (Ewl_MVC *mvc, void(*cb)(Ewl_MVC *mvc))
This callback will be called whenever the ewl_mvc_view_set routine is called to notify the inheriting widget that the view has changed.
- Ewl_View * **ewl_mvc_view_get** (Ewl_MVC *mvc)
Retrives the current view set on the MVC.

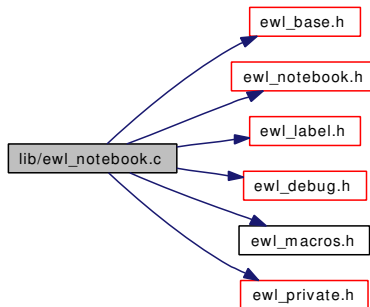
- void ewl_mvc_view_set (Ewl_MVC *mvc, Ewl_View *view)

Sets the given view onto the MVC.

8.145 lib/ewl_notebook.c File Reference

```
#include "ewl_base.h"
#include "ewl_notebook.h"
#include "ewl_label.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_notebook.c:



Functions

- void **ewl_notebook_cb_child_add** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_notebook_cb_child_hide** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_notebook_cb_child_remove** (Ewl_Container *c, Ewl_Widget *w, int rem_idx __UNUSED__)
- void **ewl_notebook_cb_child_show** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_notebook_cb_tab_clicked** (Ewl_Widget *w, void *ev __UNUSED__, void *data)
- int **ewl_notebook_init** (Ewl_Notebook *n)

Initialize a notebook to default values and callbacks.
- Ewl_Widget * **ewl_notebook_new** (void)

Create a new notebook widget.
- const char * **ewl_notebook_page_tab_text_get** (Ewl_Notebook *n, Ewl_Widget *page)

Get the text of the notebook page page.
- void **ewl_notebook_page_tab_text_set** (Ewl_Notebook *n, Ewl_Widget *page, const char *text)

Set the text of the tab for the page page to the text text.
- Ewl_Widget * **ewl_notebook_page_tab_widget_get** (Ewl_Notebook *n, Ewl_Widget *page)

Retrieve the widget used as the tab for the page page.

- void **ewl_notebook_page_tab_widget_set** (Ewl_Notebook *n, Ewl_Widget *page, Ewl_Widget *tab)
Set the widget to use as the tab for the page `page` to widget `tab`.
- unsigned int **ewl_notebook_tabbar_alignment_get** (Ewl_Notebook *n)
Retrieves the alignment of the tabbar in the notebook widget.
- void **ewl_notebook_tabbar_alignment_set** (Ewl_Notebook *n, unsigned int align)
Set the alignment of the tabbar in the notebook widget.
- Ewl_Position **ewl_notebook_tabbar_position_get** (Ewl_Notebook *n)
Get the position of the tabbar in the notebook widget.
- void **ewl_notebook_tabbar_position_set** (Ewl_Notebook *n, Ewl_Position pos)
Set the position of the tabbar in the notebook widget.
- unsigned int **ewl_notebook_tabbar_visible_get** (Ewl_Notebook *n)
Retrieve if the tabbar is visible.
- void **ewl_notebook_tabbar_visible_set** (Ewl_Notebook *n, unsigned int visible)
Set if the tabbar is visible.
- Ewl_Widget * **ewl_notebook_visible_page_get** (Ewl_Notebook *n)
Get the current visible page of the notebook.
- void **ewl_notebook_visible_page_set** (Ewl_Notebook *n, Ewl_Widget *page)
Set the current visible page of the notebook.

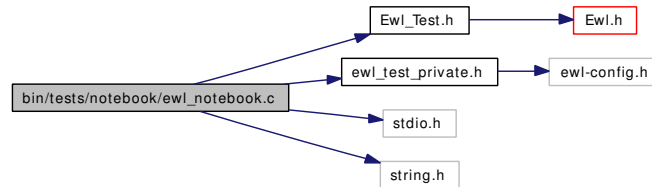
8.145.1 Function Documentation

- 8.145.1.1 void **ewl_notebook_cb_tab_clicked** (Ewl_Widget * *w*, void **ev*
 __ *UNUSED* __, void * *data*)

8.146 bin/tests/notebook/ewl_notebook.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <string.h>
```

Include dependency graph for ewl_notebook.c:



Functions

- `void test_info (Ewl_Test *test)`

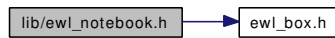
8.146.1 Function Documentation

8.146.1.1 `void test_info (Ewl_Test * test)`

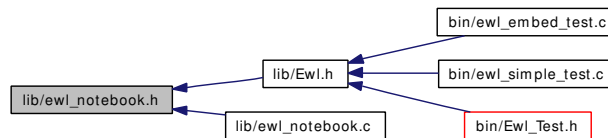
8.147 lib/ewl_notebook.h File Reference

```
#include "ewl_box.h"
```

Include dependency graph for ewl_notebook.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Notebook**

*Inherits from **Ewl_Box** (p. 586) and extends to provide for a notebook style layout.*

Defines

- `#define EWL_NOTEBOOK(n) ((Ewl_Notebook *)n)`
- `#define EWL_NOTEBOOK_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_NOTEBOOK_TYPE))`
- `#define EWL_NOTEBOOK_TYPE "notebook"`

Typedefs

- `typedef Ewl_Notebook Ewl_Notebook`

Functions

- `void ewl_notebook_cb_child_add (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_notebook_cb_child_hide (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_notebook_cb_child_remove (Ewl_Container *c, Ewl_Widget *w, int rem_idx)`
- `void ewl_notebook_cb_child_show (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_notebook_cb_tab_clicked (Ewl_Widget *w, void *ev, void *data)`
- `int ewl_notebook_init (Ewl_Notebook *n)`

Initialize a notebook to default values and callbacks.

- `Ewl_Widget * ewl_notebook_new (void)`

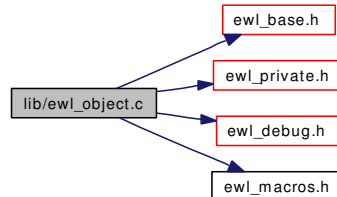
Create a new notebook widget.

- `const char * ewl_notebook_page_tab_text_get (Ewl_Notebook *n, Ewl_Widget *page)`
Get the text of the notebook page `page`.
- `void ewl_notebook_page_tab_text_set (Ewl_Notebook *n, Ewl_Widget *page, const char *text)`
Set the text of the tab for the page `page` to the text `text`.
- `Ewl_Widget * ewl_notebook_page_tab_widget_get (Ewl_Notebook *n, Ewl_Widget *page)`
Retrieve the widget used as the tab for the page `page`.
- `void ewl_notebook_page_tab_widget_set (Ewl_Notebook *n, Ewl_Widget *page, Ewl_Widget *tab)`
Set the widget to use as the tab for the page `page` to widget `tab`.
- `unsigned int ewl_notebook_tabbar_alignment_get (Ewl_Notebook *n)`
Retrieves the alignment of the tabbar in the notebook widget.
- `void ewl_notebook_tabbar_alignment_set (Ewl_Notebook *n, unsigned int align)`
Set the alignment of the tabbar in the notebook widget.
- `Ewl_Position ewl_notebook_tabbar_position_get (Ewl_Notebook *n)`
Get the position of the tabbar in the notebook widget.
- `void ewl_notebook_tabbar_position_set (Ewl_Notebook *n, Ewl_Position pos)`
Set the position of the tabbar in the notebook widget.
- `unsigned int ewl_notebook_tabbar_visible_get (Ewl_Notebook *n)`
Retrieve if the tabbar is visible.
- `void ewl_notebook_tabbar_visible_set (Ewl_Notebook *n, unsigned int visible)`
Set if the tabbar is visible.
- `Ewl_Widget * ewl_notebook_visible_page_get (Ewl_Notebook *n)`
Get the current visible page of the notebook.
- `void ewl_notebook_visible_page_set (Ewl_Notebook *n, Ewl_Widget *page)`
Set the current visible page of the notebook.

8.148 lib/ewl_object.c File Reference

```
#include "ewl_base.h"
#include "ewl_private.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
```

Include dependency graph for ewl_object.c:



Functions

- unsigned int **ewl_object_alignment_get** (Ewl_Object *o)
Retrieves the current setting of the object alignment for o.
- void **ewl_object_alignment_set** (Ewl_Object *o, unsigned int align)
Change the alignment of the specified object.
- void **ewl_object_current_geometry_get** (Ewl_Object *o, int *x, int *y, int *w, int *h)
Retrieve the size and position of object.
- int **ewl_object_current_h_get** (Ewl_Object *o)
Get the current height of the object.
- void **ewl_object_current_size_get** (Ewl_Object *o, int *w, int *h)
Retrieve the current size of an object.
- int **ewl_object_current_w_get** (Ewl_Object *o)
Get the current width of the object.
- int **ewl_object_current_x_get** (Ewl_Object *o)
get the current x position of the object
- int **ewl_object_current_y_get** (Ewl_Object *o)
Get the current y position of the object.
- unsigned int **ewl_object_fill_policy_get** (Ewl_Object *o)
Retrieves the current setting of the object fill policy for o.
- void **ewl_object_fill_policy_set** (Ewl_Object *o, unsigned int fill)
Change the fill policy of the specified object.

- void **ewl_object_flags_add** (Ewl_Object *o, unsigned int flags, unsigned int mask)
Add the set of flags specified in flags to o.
- void **ewl_object_flags_remove** (Ewl_Object *o, unsigned int flags, unsigned int mask)
Removes the set of state flags specified in flags from o.
- void **ewl_object_geometry_request** (Ewl_Object *o, int x, int y, int w, int h)
Request a new geometry be applied to an object.
- void **ewl_object_h_request** (Ewl_Object *o, int h)
Request a new width for an object.
- int **ewl_object_init** (Ewl_Object *o)
- int **ewl_object_insets_bottom_get** (Ewl_Object *o)
Get the bottom insets for the object.
- void **ewl_object_insets_get** (Ewl_Object *o, int *l, int *r, int *t, int *b)
Retrieve the edge insets of an object.
- int **ewl_object_insets_left_get** (Ewl_Object *o)
Get the left insets for the object.
- int **ewl_object_insets_right_get** (Ewl_Object *o)
Get the right insets for the object.
- void **ewl_object_insets_set** (Ewl_Object *o, int l, int r, int t, int b)
Set the insets around an objects edges.
- int **ewl_object_insets_top_get** (Ewl_Object *o)
Get the top insets for the object.
- int **ewl_object_maximum_h_get** (Ewl_Object *o)
Get the maximum height of an object.
- void **ewl_object_maximum_h_set** (Ewl_Object *o, int h)
Set the minimum height of an object.
- int **ewl_object_maximum_inner_h_get** (Ewl_Object *o)
Get the maximum height of an object.
- int **ewl_object_maximum_inner_w_get** (Ewl_Object *o)
Get the maximum width of an object.
- void **ewl_object_maximum_size_get** (Ewl_Object *o, int *w, int *h)
Retrieve the minimum dimensions of an object.
- void **ewl_object_maximum_size_set** (Ewl_Object *o, int w, int h)
Set the maximum size of an object.

- `int ewl_object_maximum_w_get (Ewl_Object *o)`
Get the maximum width of an object.
- `void ewl_object_maximum_w_set (Ewl_Object *o, int w)`
Set the minimum width of an object.
- `int ewl_object_minimum_h_get (Ewl_Object *o)`
Get the minimum height of an object.
- `void ewl_object_minimum_h_set (Ewl_Object *o, int h)`
Set the minimum height of an object.
- `int ewl_object_minimum_inner_h_get (Ewl_Object *o)`
Get the minimum height of an object.
- `int ewl_object_minimum_inner_w_get (Ewl_Object *o)`
Get the minimum width of an object.
- `void ewl_object_minimum_size_get (Ewl_Object *o, int *w, int *h)`
Retrieve the minimum dimensions of an object.
- `void ewl_object_minimum_size_set (Ewl_Object *o, int w, int h)`
Set the minimum size of an object.
- `int ewl_object_minimum_w_get (Ewl_Object *o)`
Get the minimum width of an object.
- `void ewl_object_minimum_w_set (Ewl_Object *o, int w)`
Set the minimum width of an object.
- `int ewl_object_padding_bottom_get (Ewl_Object *o)`
Get the bottom padding for the object.
- `void ewl_object_padding_get (Ewl_Object *o, int *l, int *r, int *t, int *b)`
Retrieve the edge padding of an object.
- `int ewl_object_padding_left_get (Ewl_Object *o)`
Get the left padding for the object.
- `int ewl_object_padding_right_get (Ewl_Object *o)`
Get the right padding for the object.
- `void ewl_object_padding_set (Ewl_Object *o, int l, int r, int t, int b)`
Set the padding around an objects edges.
- `int ewl_object_padding_top_get (Ewl_Object *o)`
Get the top padding for the object.
- `void ewl_object_place (Ewl_Object *o, int x, int y, int w, int h)`

Assign a specific area to an object.

- **void ewl_object_position_request** (Ewl_Object *o, int x, int y)
Request a new position be applied to an object.
- **int ewl_object_preferred_h_get** (Ewl_Object *o)
Get the preferred height of the object.
- **int ewl_object_preferred_inner_h_get** (Ewl_Object *o)
Get the preferred height of the object.
- **void ewl_object_preferred_inner_h_set** (Ewl_Object *o, int h)
Set the preferred height of the object.
- **void ewl_object_preferred_inner_size_get** (Ewl_Object *o, int *w, int *h)
Retrieve the preferred size of an object.
- **void ewl_object_preferred_inner_size_set** (Ewl_Object *o, int w, int h)
Set the preferred size of the object.
- **int ewl_object_preferred_inner_w_get** (Ewl_Object *o)
Get the preferred width of the object, ignoring padding and insets.
- **void ewl_object_preferred_inner_w_set** (Ewl_Object *o, int w)
Set the preferred width of the object.
- **void ewl_object_preferred_size_get** (Ewl_Object *o, int *w, int *h)
Retrieve the preferred size of an object.
- **int ewl_object_preferred_w_get** (Ewl_Object *o)
Get the preferred width of the object.
- **void ewl_object_size_request** (Ewl_Object *o, int w, int h)
Request a new size be applied to an object.
- **void ewl_object_w_request** (Ewl_Object *o, int w)
Request a new width for an object.
- **void ewl_object_x_request** (Ewl_Object *o, int x)
Request a new x position for an object.
- **void ewl_object_y_request** (Ewl_Object *o, int y)
Request a new y position for an object.

8.148.1 Function Documentation

8.148.1.1 int ewl_object_maximum_inner_h_get (Ewl_Object * o)

Get the maximum height of an object.

Parameters:

o,: the object to get the maximum height

Returns:

Returns the maximum height of the object.

8.148.1.2 int ewl_object_maximum_inner_w_get (Ewl_Object * o)

Get the maximum width of an object.

Parameters:

o,: the object to get the maximum width

Returns:

Returns the maximum width of the object.

8.148.1.3 int ewl_object_minimum_inner_h_get (Ewl_Object * o)

Get the minimum height of an object.

Parameters:

o,: the object to get the minimum height

Returns:

Returns the minimum height of the object.

8.148.1.4 int ewl_object_minimum_inner_w_get (Ewl_Object * o)

Get the minimum width of an object.

Parameters:

o,: the object to get the minimum width

Returns:

Returns the minimum width of the object *o*.

8.149 lib/ewl_object.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Object**

Provides facilities for sizing, position, alignment and fill policy.

Defines

- `#define CONFIGURED(o)`
- `#define CURRENT_H(o) EWL_OBJECT(o) → current.h`
- `#define CURRENT_W(o) EWL_OBJECT(o) → current.w`
- `#define CURRENT_X(o) EWL_OBJECT(o) → current.x`
- `#define CURRENT_Y(o) EWL_OBJECT(o) → current.y`
- `#define DESTROYED(o)`
- `#define DISABLED(o) (ewl_object_state_has(EWL_OBJECT(o), EWL_FLAG_STATE_DISABLED))`
- `#define EWL_OBJECT(object) ((Ewl_Object *) object)`
- `#define ewl_object_custom_h_set(o, h)`
- `#define ewl_object_custom_size_set(o, w, h)`
- `#define ewl_object_custom_w_set(o, w)`
- `#define ewl_object_flags_get(o, mask) (EWL_OBJECT(o) → flags & mask)`
Retrieves the current setting of the object flags for o.
- `#define ewl_object_flags_has(o, check_flags, mask) (!(EWL_OBJECT(o) → flags & ((check_flags) & mask)))`
Determines if an object has the requested flags set.
- `#define ewl_object_flags_has_all(o, check_flags, mask) ((EWL_OBJECT(o) → flags & ((check_flags) & mask)) == ((check_flags) & mask))`
Determines if an object has all of the requested flags set.
- `#define ewl_object_in_tab_list_get(o) (ewl_object_flags_get(o, EWL_FLAG_PROPERTY_IN_TAB_LIST))`
Retrieves the current setting of the in tab list flag for o.
- `#define ewl_object_in_tab_list_set(o, val)`
Changes the tab list flag value to match val.
- `#define EWL_OBJECT_MAX_SIZE (INT_MAX)`
- `#define EWL_OBJECT_MIN_SIZE (1)`
- `#define ewl_object_queued_add(o, queued) ewl_object_flags_add(o, queued, EWL_FLAGS_QUEUED_MASK)`

- `#define ewl_object_queued_get(o, queued) ewl_object_flags_get(o, queued, EWL_FLAGS_QUEUED_MASK)`
- `#define ewl_object_queued_has(o, queued) ewl_object_flags_has(o, queued, EWL_FLAGS_QUEUED_MASK)`
- `#define ewl_object_queued_remove(o, queued) ewl_object_flags_remove(o, queued, EWL_FLAGS_QUEUED_MASK)`
- `#define ewl_object_recursive_get(o) (ewl_object_flags_get(o, EWL_FLAG_PROPERTY_RECURSIVE))`

Retrieves the current setting of the recursive flag for o.

- `#define ewl_object_recursive_set(o, val)`

Changes the recursive flag value to match val.

- `#define ewl_object_state_add(o, state) ewl_object_flags_add(o, state, EWL_FLAGS_STATE_MASK)`
- `#define ewl_object_state_get(o, state) ewl_object_flags_get(o, state, EWL_FLAGS_STATE_MASK)`
- `#define ewl_object_state_has(o, state) ewl_object_flags_has(o, state, EWL_FLAGS_STATE_MASK)`
- `#define ewl_object_state_remove(o, state) ewl_object_flags_remove(o, state, EWL_FLAGS_STATE_MASK)`
- `#define ewl_object_toplevel_get(o) (ewl_object_flags_get(o, EWL_FLAG_PROPERTY_TOPLEVEL))`

Retrieves the current setting of the top level flag for o.

- `#define ewl_object_toplevel_set(o, val)`

Changes the top level flag value to match val.

- `#define ewl_object_visible_add(o, visible) ewl_object_flags_add(o, visible, EWL_FLAGS_VISIBLE_MASK)`
- `#define ewl_object_visible_get(o, visible) ewl_object_flags_get(o, visible, EWL_FLAGS_VISIBLE_MASK)`
- `#define ewl_object_visible_has(o, visible) ewl_object_flags_has(o, visible, EWL_FLAGS_VISIBLE_MASK)`
- `#define ewl_object_visible_remove(o, visible) ewl_object_flags_remove(o, visible, EWL_FLAGS_VISIBLE_MASK)`
- `#define HIDDEN(o) (!(EWL_OBJECT(o) → flags & EWL_FLAG_VISIBLE_SHOWN))`
- `#define INSET_BOTTOM(o) EWL_OBJECT(o) → insets.b`
- `#define INSET_HORIZONTAL(o) (EWL_OBJECT(o) → insets.l + EWL_OBJECT(o) → insets.r)`
- `#define INSET_LEFT(o) EWL_OBJECT(o) → insets.l`
- `#define INSET_RIGHT(o) EWL_OBJECT(o) → insets.r`
- `#define INSET_TOP(o) EWL_OBJECT(o) → insets.t`
- `#define INSET_VERTICAL(o) (EWL_OBJECT(o) → insets.t + EWL_OBJECT(o) → insets.b)`
- `#define MAXIMUM_H(o) EWL_OBJECT(o) → maximum.h`
- `#define MAXIMUM_W(o) EWL_OBJECT(o) → maximum.w`
- `#define MINIMUM_H(o) EWL_OBJECT(o) → minimum.h`
- `#define MINIMUM_W(o) EWL_OBJECT(o) → minimum.w`

- `#define OBSCURED(o) (EWL_OBJECT(o) → flags & EWL_FLAG_VISIBLE_-OBSCURED)`
- `#define PADDING_BOTTOM(o) EWL_OBJECT(o) → pad.b`
- `#define PADDING_HORIZONTAL(o) (EWL_OBJECT(o) → pad.l + EWL_OBJECT(o) → pad.r)`
- `#define PADDING_LEFT(o) EWL_OBJECT(o) → pad.l`
- `#define PADDING_RIGHT(o) EWL_OBJECT(o) → pad.r`
- `#define PADDING_TOP(o) EWL_OBJECT(o) → pad.t`
- `#define PADDING_VERTICAL(o) (EWL_OBJECT(o) → pad.t + EWL_OBJECT(o) → pad.b)`
- `#define PREFERRED_H(o) EWL_OBJECT(o) → preferred.h`
- `#define PREFERRED_W(o) EWL_OBJECT(o) → preferred.w`
- `#define REALIZED(o) (EWL_OBJECT(o) → flags & EWL_FLAG_VISIBLE_-REALIZED)`
- `#define RECURSIVE(o) (EWL_OBJECT(o) → flags & EWL_FLAG_PROPERTY_-RECURSIVE)`
- `#define VISIBLE(o) (EWL_OBJECT(o) → flags & EWL_FLAG_VISIBLE_SHOWN)`

Typedefs

- `typedef Ewl_Object Ewl_Object`

Functions

- `unsigned int ewl_object_alignment_get (Ewl_Object *o)`
Retrieves the current setting of the object alignment for o.
- `void ewl_object_alignment_set (Ewl_Object *o, unsigned int align)`
Change the alignment of the specified object.
- `void ewl_object_current_geometry_get (Ewl_Object *o, int *x, int *y, int *w, int *h)`
Retrieve the size and position of object.
- `int ewl_object_current_h_get (Ewl_Object *o)`
Get the current height of the object.
- `void ewl_object_current_size_get (Ewl_Object *o, int *w, int *h)`
Retrieve the current size of an object.
- `int ewl_object_current_w_get (Ewl_Object *o)`
Get the current width of the object.
- `int ewl_object_current_x_get (Ewl_Object *o)`
get the current x position of the object
- `int ewl_object_current_y_get (Ewl_Object *o)`
Get the current y position of the object.
- `unsigned int ewl_object_fill_policy_get (Ewl_Object *o)`

Retrieves the current setting of the object fill policy for o.

- void **ewl_object_fill_policy_set** (Ewl_Object *o, unsigned int fill)
Change the fill policy of the specified object.
- void **ewl_object_flags_add** (Ewl_Object *o, unsigned int flags, unsigned int mask)
Add the set of flags specified in flags to o.
- void **ewl_object_flags_remove** (Ewl_Object *o, unsigned int flags, unsigned int mask)
Removes the set of state flags specified in flags from o.
- void **ewl_object_geometry_request** (Ewl_Object *o, int x, int y, int w, int h)
Request a new geometry be applied to an object.
- void **ewl_object_h_request** (Ewl_Object *o, int h)
Request a new width for an object.
- int **ewl_object_init** (Ewl_Object *o)
- int **ewl_object_insets_bottom_get** (Ewl_Object *o)
Get the bottom insets for the object.
- void **ewl_object_insets_get** (Ewl_Object *o, int *l, int *r, int *t, int *b)
Retrieve the edge insets of an object.
- int **ewl_object_insets_left_get** (Ewl_Object *o)
Get the left insets for the object.
- int **ewl_object_insets_right_get** (Ewl_Object *o)
Get the right insets for the object.
- void **ewl_object_insets_set** (Ewl_Object *o, int l, int r, int t, int b)
Set the insets around an objects edges.
- int **ewl_object_insets_top_get** (Ewl_Object *o)
Get the top insets for the object.
- int **ewl_object_maximum_h_get** (Ewl_Object *o)
Get the maximum height of an object.
- void **ewl_object_maximum_h_set** (Ewl_Object *o, int h)
Set the minimum height of an object.
- void **ewl_object_maximum_size_get** (Ewl_Object *o, int *w, int *h)
Retrieve the minimum dimensions of an object.
- void **ewl_object_maximum_size_set** (Ewl_Object *o, int w, int h)
Set the maximum size of an object.
- int **ewl_object_maximum_w_get** (Ewl_Object *o)

Get the maximum width of an object.

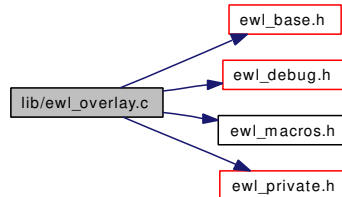
- void **ewl_object_maximum_w_set** (Ewl_Object *o, int w)
Set the minimum width of an object.
- int **ewl_object_minimum_h_get** (Ewl_Object *o)
Get the minimum height of an object.
- void **ewl_object_minimum_h_set** (Ewl_Object *o, int h)
Set the minimum height of an object.
- void **ewl_object_minimum_size_get** (Ewl_Object *o, int *w, int *h)
Retrieve the minimum dimensions of an object.
- void **ewl_object_minimum_size_set** (Ewl_Object *o, int w, int h)
Set the minimum size of an object.
- int **ewl_object_minimum_w_get** (Ewl_Object *o)
Get the minimum width of an object.
- void **ewl_object_minimum_w_set** (Ewl_Object *o, int w)
Set the minimum width of an object.
- int **ewl_object_padding_bottom_get** (Ewl_Object *o)
Get the bottom padding for the object.
- void **ewl_object_padding_get** (Ewl_Object *o, int *l, int *r, int *t, int *b)
Retrieve the edge padding of an object.
- int **ewl_object_padding_left_get** (Ewl_Object *o)
Get the left padding for the object.
- int **ewl_object_padding_right_get** (Ewl_Object *o)
Get the right padding for the object.
- void **ewl_object_padding_set** (Ewl_Object *o, int l, int r, int t, int b)
Set the padding around an objects edges.
- int **ewl_object_padding_top_get** (Ewl_Object *o)
Get the top padding for the object.
- void **ewl_object_place** (Ewl_Object *o, int x, int y, int w, int h)
Assign a specific area to an object.
- void **ewl_object_position_request** (Ewl_Object *o, int x, int y)
Request a new position be applied to an object.
- int **ewl_object_preferred_h_get** (Ewl_Object *o)
Get the preferred height of the object.

- **int ewl_object_preferred_inner_h_get (Ewl_Object *o)**
Get the preferred height of the object.
- **void ewl_object_preferred_inner_h_set (Ewl_Object *o, int h)**
Set the preferred height of the object.
- **void ewl_object_preferred_inner_size_get (Ewl_Object *o, int *w, int *h)**
Retrieve the preferred size of an object.
- **void ewl_object_preferred_inner_size_set (Ewl_Object *o, int w, int h)**
Set the preferred size of the object.
- **int ewl_object_preferred_inner_w_get (Ewl_Object *o)**
Get the preferred width of the object, ignoring padding and insets.
- **void ewl_object_preferred_inner_w_set (Ewl_Object *o, int w)**
Set the preferred width of the object.
- **void ewl_object_preferred_size_get (Ewl_Object *o, int *w, int *h)**
Retrieve the preferred size of an object.
- **int ewl_object_preferred_w_get (Ewl_Object *o)**
Get the preferred width of the object.
- **void ewl_object_size_request (Ewl_Object *o, int w, int h)**
Request a new size be applied to an object.
- **void ewl_object_w_request (Ewl_Object *o, int w)**
Request a new width for an object.
- **void ewl_object_x_request (Ewl_Object *o, int x)**
Request a new x position for an object.
- **void ewl_object_y_request (Ewl_Object *o, int y)**
Request a new y position for an object.

8.150 lib/ewl_overlay.c File Reference

```
#include "ewl_base.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_overlay.c:



Functions

- void **ewl_overlay_cb_child_resize** (Ewl_Container *c, Ewl_Widget *w, int size __UNUSED__, Ewl_Orientation o __UNUSED__)
- void **ewl_overlay_cb_child_show** (Ewl_Container *o, Ewl_Widget *child)
- void **ewl_overlay_cb_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- int **ewl_overlay_init** (Ewl_Overlay *w)
initialize a overlay to default values and callbacks
- Ewl_Widget * **ewl_overlay_new** (void)
Allocate and initialize a new overlay container.

8.150.1 Function Documentation

8.150.1.1 void **ewl_overlay_cb_configure** (Ewl_Widget * w, void *ev_data __UNUSED__, void *user_data __UNUSED__)

8.151 lib/ewl_overlay.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Overlay**

*The class inheriting from **Ewl_Container** (p. 618) that acts as a top level widget for interacting with the evas.*

Defines

- `#define EWL_OVERLAY(widget) ((Ewl_Overlay *) widget)`
*Typecast a pointer to an **Ewl_Overlay** (p. 712) pointer.*
- `#define EWL_OVERLAY_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_OVERLAY_TYPE))`
- `#define EWL_OVERLAY_TYPE "overlay"`

Typedefs

- `typedef Ewl_Overlay Ewl_Overlay`

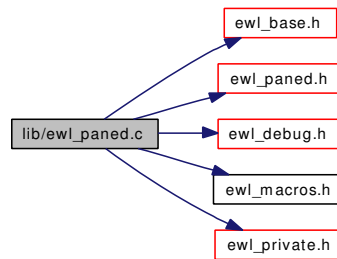
Functions

- `void ewl_overlay_cb_child_resize (Ewl_Container *c, Ewl_Widget *w, int size, Ewl_Orientation o)`
- `void ewl_overlay_cb_child_show (Ewl_Container *emb, Ewl_Widget *child)`
- `void ewl_overlay_cb_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `int ewl_overlay_init (Ewl_Overlay *win)`
initialize a overlay to default values and callbacks
- `Ewl_Widget * ewl_overlay_new (void)`
Allocate and initialize a new overlay container.

8.152 lib/ewl_paned.c File Reference

```
#include "ewl_base.h"
#include "ewl_paned.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_paned.c:



Data Structures

- struct **Ewl_Paned_Layout**
Contains information on a paned layout.

Functions

- **Ewl_Widget * ewl_hpaned_new** (void)
Allocate and initialize a new paned widget with horizontal orientation.
- void **ewl_paned_cb_child_add** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_paned_cb_child_hide** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_paned_cb_child_remove** (Ewl_Container *c, Ewl_Widget *w, int idx)
- void **ewl_paned_cb_child_show** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_paned_cb_configure** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- int **ewl_paned_grabber_init** (Ewl_Paned_Grabber *g)
- **Ewl_Widget * ewl_paned_grabber_new** (void)
- **Ewl_Orientation ewl_paned_grabber_orientation_get** (Ewl_Paned_Grabber *g)
- void **ewl_paned_grabber_paned_orientation_set** (Ewl_Paned_Grabber *g, Ewl_Orientation o)
- void **ewl_paned_grabber_show_cursor_for** (Ewl_Paned_Grabber *g, unsigned int dir)
- int **ewl_paned_init** (Ewl_Paned *p)
Initialize a new paned widget to default values.
- **Ewl_Widget * ewl_paned_new** (void)

Allocate and initialize a new paned widget.

- **Ewl_Orientation ewl_paned_orientation_get (Ewl_Paned *p)**
Get the orientation of the paned widget.
- **void ewl_paned_orientation_set (Ewl_Paned *p, Ewl_Orientation o)**
Set the orientation of the paned.
- **Ewl_Widget * ewl_vpaned_new (void)**
Allocate and initialize a new paned widget with vertical orientation.

8.152.1 Function Documentation

**8.152.1.1 void ewl_paned_cb_configure (Ewl_Widget * w, void *ev
__ *UNUSED* __, void *data __ *UNUSED* __)**

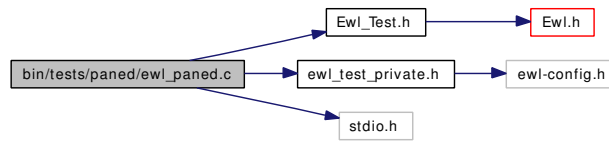
8.153 bin/tests/paned/ewl_paned.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_paned.c:



Functions

- `void test_info (Ewl_Test *test)`

8.153.1 Function Documentation

8.153.1.1 `void test_info (Ewl_Test * test)`

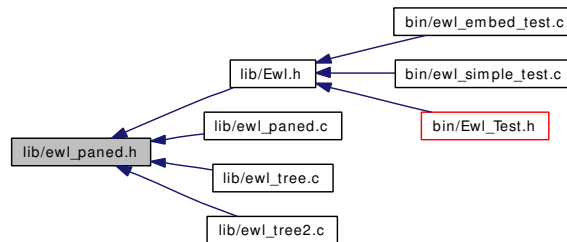
8.154 lib/ewl_paned.h File Reference

```
#include "ewl_separator.h"
```

Include dependency graph for ewl_paned.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Paned**
*Inherits from **Ewl_Container** (p. 618) and extends to provided the paned widget.*
- struct **Ewl_Paned_Grabber**
*Inherits from **Ewl_Separator** (p. 740) and extends to facilitate the paned grabber.*

Defines

- `#define EWL_PANED(paned) ((Ewl_Paned *) paned)`
- `#define EWL_PANED_GRABBER(g) ((Ewl_Paned_Grabber *)g)`
- `#define EWL_PANED_GRABBER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_PANED_GRABBER_TYPE))`
- `#define EWL_PANED_GRABBER_TYPE "paned_grabber"`
- `#define EWL_PANED_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_PANED_TYPE))`
- `#define EWL_PANED_TYPE "paned"`

Typedefs

- `typedef Ewl_Paned Ewl_Paned`
- `typedef Ewl_Paned_Grabber Ewl_Paned_Grabber`

Functions

- `Ewl_Widget * ewl_hpaned_new (void)`
Allocate and initialize a new paned widget with horizontal orientation.

- void ewl_paned_cb_child_add (Ewl_Container *c, Ewl_Widget *w)
- void ewl_paned_cb_child_hide (Ewl_Container *c, Ewl_Widget *w)
- void ewl_paned_cb_child_remove (Ewl_Container *c, Ewl_Widget *w, int idx)
- void ewl_paned_cb_child_show (Ewl_Container *c, Ewl_Widget *w)
- void ewl_paned_cb_configure (Ewl_Widget *w, void *ev, void *data)
- void ewl_paned_cb_destroy (Ewl_Widget *w, void *ev, void *data)
- int ewl_paned_grabber_init (Ewl_Paned_Grabber *g)
- Ewl_Widget * ewl_paned_grabber_new (void)
- Ewl_Orientation ewl_paned_grabber_orientation_get (Ewl_Paned_Grabber *g)
- void ewl_paned_grabber_paned_orientation_set (Ewl_Paned_Grabber *g, Ewl_Orientation o)
- void ewl_paned_grabber_show_cursor_for (Ewl_Paned_Grabber *g, unsigned int dir)
- int ewl_paned_init (Ewl_Paned *p)

Initialize a new paned widget to default values.

- Ewl_Widget * ewl_paned_new (void)

Allocate and initialize a new paned widget.

- Ewl_Orientation ewl_paned_orientation_get (Ewl_Paned *p)

Get the orientation of the paned widget.

- void ewl_paned_orientation_set (Ewl_Paned *p, Ewl_Orientation o)

Set the orientation of the paned.

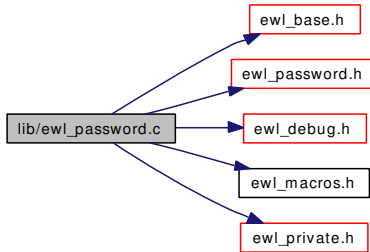
- Ewl_Widget * ewl_vpaned_new (void)

Allocate and initialize a new paned widget with vertical orientation.

8.155 lib/ewl_password.c File Reference

```
#include "ewl_base.h"
#include "ewl_password.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_password.c:



Functions

- void **ewl_password_cb_destroy** (**Ewl_Widget** *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_password_cb_key_down** (**Ewl_Widget** *w, void *ev_data, void *user_data __UNUSED__)
- int **ewl_password_init** (**Ewl_Password** *e)

Initialize an password widget to default values.
- **Ewl_Widget** * **ewl_password_new** (void)

Allocate and initialize a new password widget.
- char **ewl_password_obscure_get** (**Ewl_Password** *e)

Retrieves the character used to obscure the text for a password.
- void **ewl_password_obscure_set** (**Ewl_Password** *e, char o)

Sets the character used to obscure the text for a password.
- char * **ewl_password_text_get** (**Ewl_Password** *e)

Get the text from an password widget.
- void **ewl_password_text_set** (**Ewl_Password** *e, const char *t)

Set the text for an password widget.

8.155.1 Function Documentation

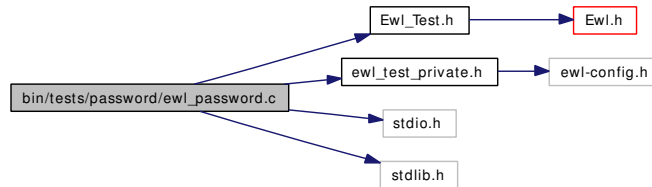
8.155.1.1 void ewl_password_cb_destroy (Ewl_Widget * *w*, void *ev_data __ *UNUSED* __, void *user_data __ *UNUSED* __)

8.155.1.2 void ewl_password_cb_key_down (Ewl_Widget * *w*, void * *ev_data*, void *user_data __ *UNUSED* __)

8.156 bin/tests/password/ewl_password.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <stdlib.h>
```

Include dependency graph for ewl_password.c:



Functions

- void `test_info` (Ewl_Test *test)

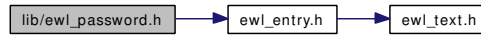
8.156.1 Function Documentation

8.156.1.1 void `test_info` (Ewl_Test * *test*)

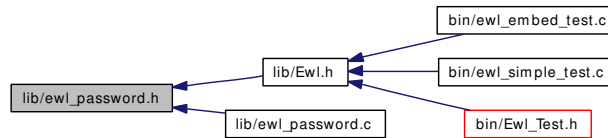
8.157 lib/ewl_password.h File Reference

```
#include "ewl_entry.h"
```

Include dependency graph for ewl_password.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Password**

*Inherits from the **Ewl_Entry** (p. 635) and extends it to obscure the text.*

Defines

- #define **EWL_PASSWORD**(password) ((**Ewl_Password** *) password)
- #define **EWL_PASSWORD_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_PASSWORD_TYPE**))
- #define **EWL_PASSWORD_TYPE** "password"

Typedefs

- typedef **Ewl_Password** **Ewl_Password**

Functions

- void **ewl_password_cb_destroy** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_password_cb_key_down** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- int **ewl_password_init** (**Ewl_Password** *e)
Initialize an password widget to default values.
- **Ewl_Widget** * **ewl_password_new** (void)
Allocate and initialize a new password widget.
- char **ewl_password_obscure_get** (**Ewl_Password** *e)
Retrieves the character used to obscure the text for a password.
- void **ewl_password_obscure_set** (**Ewl_Password** *e, char o)

Sets the character used to obscure the text for a password.

- `char * ewl_password_text_get (Ewl_Password *e)`

Get the text from an password widget.

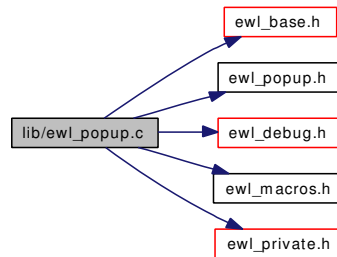
- `void ewl_password_text_set (Ewl_Password *e, const char *t)`

Set the text for an password widget.

8.158 lib/ewl_popup.c File Reference

```
#include "ewl_base.h"
#include "ewl_popup.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_popup.c:



Functions

- void **ewl_popup_cb_child_resize** (Ewl_Container *c, Ewl_Widget *w, int size, Ewl_Orientation o)
- void **ewl_popup_cb_child_show** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_popup_cb_destroy** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_popup_cb_follow_configure** (Ewl_Widget *w __UNUSED__, void *ev_data __UNUSED__, void *user_data)
- void **ewl_popup_cb_follow_destroy** (Ewl_Widget *w __UNUSED__, void *ev_data __UNUSED__, void *user_data)
- void **ewl_popup_cb_show** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- int **ewl_popup_fit_to_follow_get** (Ewl_Popup *p)
- void **ewl_popup_fit_to_follow_set** (Ewl_Popup *p, int fit)
- Ewl_Widget * **ewl_popup_follow_get** (Ewl_Popup *p)
- void **ewl_popup_follow_set** (Ewl_Popup *p, Ewl_Widget *w)

Set the follow widget of the popup.

- int **ewl_popup_init** (Ewl_Popup *p)

Initialize the popup and inherited fields.

- void **ewl_popup_mouse_position_set** (Ewl_Popup *p, int x, int y)

This is to set the mouse coordinates to the popup, where it get placed relative to.

- Ewl_Widget * **ewl_popup_new** (void)

Allocate a new popup widget.

- void **ewl_popup_offset_set** (Ewl_Popup *p, int x, int y)

This is to set the offset to where the popup will be placed.

- **Ewl_Popup_Type** ewl_popup_type_get (Ewl_Popup *p)

This is to get the type of the popup. The type defines how the popup will handle the positioning.

- void ewl_popup_type_set (Ewl_Popup *p, Ewl_Popup_Type type)

This is to set the type of the popup. The type defines how the popup will handle the positioning.

8.158.1 Function Documentation

8.158.1.1 void ewl_popup_cb_destroy (Ewl_Widget * w, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)

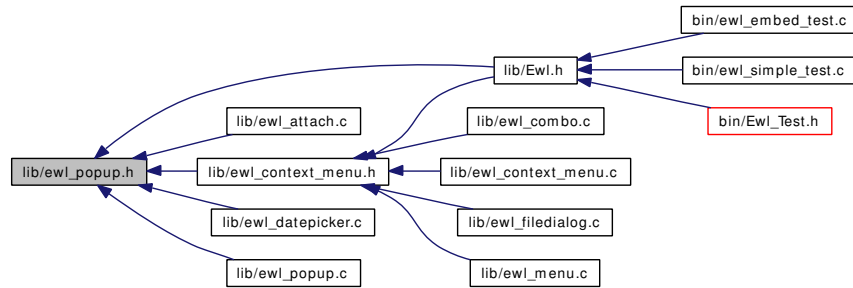
8.158.1.2 void ewl_popup_cb_follow_configure (Ewl_Widget *w
__ *UNUSED* __, void *ev_data __ *UNUSED* __, void * user_data)

8.158.1.3 void ewl_popup_cb_follow_destroy (Ewl_Widget *w __ *UNUSED* __,
void *ev_data __ *UNUSED* __, void * user_data)

8.158.1.4 void ewl_popup_cb_show (Ewl_Widget * w, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)

8.159 lib/ewl_popup.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Popup**

*Inherits from **Ewl_Window** (p. 788) and extends to provide a popup widget.*

Defines

- `#define EWL_POPUP(popup) ((Ewl_Popup *) popup)`
- `#define EWL_POPUP_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_POPUP_TYPE))`
- `#define EWL_POPUP_TYPE "popup"`

Typedefs

- `typedef Ewl_Popup Ewl_Popup`

Functions

- `void ewl_popup_cb_child_resize (Ewl_Container *c, Ewl_Widget *w, int size, Ewl_Orientation o)`
- `void ewl_popup_cb_child_show (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_popup_cb_destroy (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_popup_cb_follow_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_popup_cb_follow_destroy (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_popup_cb_show (Ewl_Widget *w, void *ev_data, void *user_data)`
- `int ewl_popup_fit_to_follow_get (Ewl_Popup *p)`
- `void ewl_popup_fit_to_follow_set (Ewl_Popup *p, int fit)`
- `Ewl_Widget * ewl_popup_follow_get (Ewl_Popup *p)`
- `void ewl_popup_follow_set (Ewl_Popup *p, Ewl_Widget *w)`

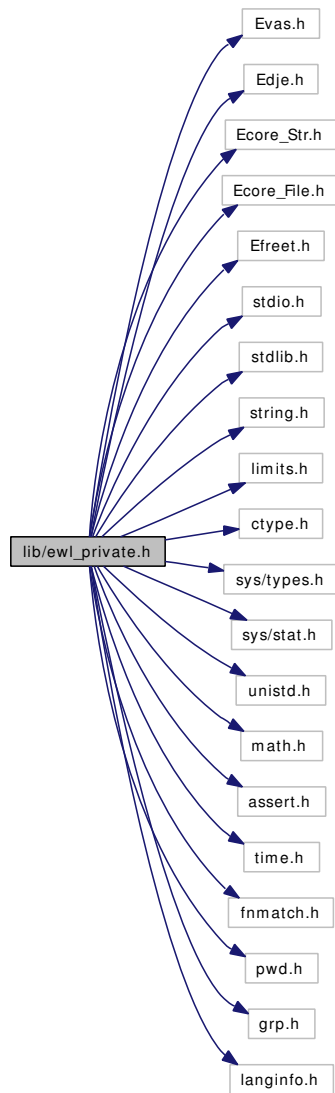
Set the follow widget of the popup.

- **int ewl_popup_init (Ewl_Popup *p)**
Initialize the popup and inherited fields.
- **void ewl_popup_mouse_position_set (Ewl_Popup *p, int x, int y)**
This is to set the mouse coordinates to the popup, where it get placed relative to.
- **Ewl_Widget * ewl_popup_new (void)**
Allocate a new popup widget.
- **void ewl_popup_offset_set (Ewl_Popup *p, int x, int y)**
This is to set the offset to where the popup will be placed.
- **Ewl_Popup_Type ewl_popup_type_get (Ewl_Popup *p)**
This is to get the type of the popup. The type defines how the popup will handle the positioning.
- **void ewl_popup_type_set (Ewl_Popup *p, Ewl_Popup_Type type)**
This is to set the type of the popup. The type defines how the popup will handle the positioning.

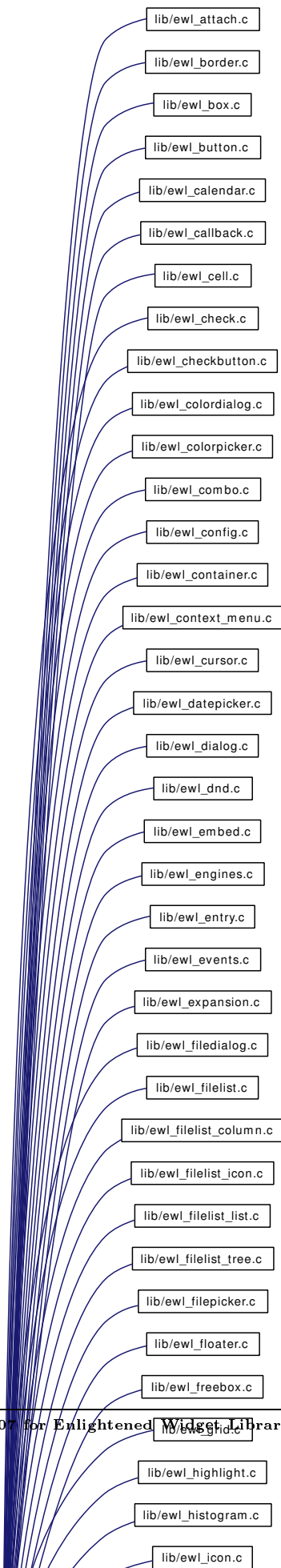
8.160 lib/ewl_private.h File Reference

```
#include <Evas.h>
#include <Edje.h>
#include <Ecore_Str.h>
#include <Ecore_File.h>
#include <Efreet.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <limits.h>
#include <ctype.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <math.h>
#include <assert.h>
#include <time.h>
#include <fnmatch.h>
#include <pwd.h>
#include <grp.h>
#include <langinfo.h>
```

Include dependency graph for ewl_private.h:



This graph shows which files directly or indirectly include this file:



Defines

- `#define __UNUSED__`
- `#define EWL_CONFIG_CACHE_EVAS_FONT "/ewl/cache/ewas/font"`
- `#define EWL_CONFIG_CACHE_EVAS_IMAGE "/ewl/cache/ewas/image"`
- `#define EWL_CONFIG_DEBUG_BACKTRACE "/ewl/debug/backtrace"`
- `#define EWL_CONFIG_DEBUG_ENABLE "/ewl/debug/enable"`
- `#define EWL_CONFIG_DEBUG_EVAS_RENDER "/ewl/debug/ewas/render"`
- `#define EWL_CONFIG_DEBUG_GC_REAP "/ewl/debug/gc/reap"`
- `#define EWL_CONFIG_DEBUG_LEVEL "/ewl/debug/level"`
- `#define EWL_CONFIG_DEBUG_SEGV "/ewl/debug/segv"`
- `#define EWL_CONFIG_ENGINE_NAME "/ewl/engine/name"`
- `#define EWL_CONFIG_THEME_COLOR_CLASSES_COUNT "/ewl/theme/color/classes/count"`
- `#define EWL_CONFIG_THEME_COLOR_CLASSES_OVERRIDE "/ewl/theme/color/classes/override"`
- `#define EWL_CONFIG_THEME_ICON_SIZE "/ewl/theme/icon/size"`
- `#define EWL_CONFIG_THEME_ICON_THEME "/ewl/theme/icon/theme"`
- `#define EWL_CONFIG_THEME_NAME "/ewl/theme/name"`
- `#define EWL_CONFIG_THEME_PRINT_KEYS "/ewl/theme/print/keys"`
- `#define EWL_CONFIG_THEME_PRINT_SIGNALS "/ewl/theme/print/signals"`
- `#define PATH_MAX 4096`

Variables

- `Ewl_Config * ewl_config`

8.160.1 Define Documentation

- 8.160.1.1 `#define __UNUSED __`
- 8.160.1.2 `#define EWL_CONFIG_CACHE_EVAS_FONT "/ewl/cache/evas/font"`
- 8.160.1.3 `#define EWL_CONFIG_CACHE_EVAS_IMAGE "/ewl/cache/evas/image"`
- 8.160.1.4 `#define EWL_CONFIG_DEBUG_BACKTRACE "/ewl/debug/backtrace"`
- 8.160.1.5 `#define EWL_CONFIG_DEBUG_ENABLE "/ewl/debug/enable"`
- 8.160.1.6 `#define EWL_CONFIG_DEBUG_EVAS_RENDER "/ewl/debug/evas/render"`
- 8.160.1.7 `#define EWL_CONFIG_DEBUG_GC_REAP "/ewl/debug/gc/reap"`
- 8.160.1.8 `#define EWL_CONFIG_DEBUG_LEVEL "/ewl/debug/level"`
- 8.160.1.9 `#define EWL_CONFIG_DEBUG_SEGV "/ewl/debug/segv"`
- 8.160.1.10 `#define EWL_CONFIG_ENGINE_NAME "/ewl/engine/name"`
- 8.160.1.11 `#define EWL_CONFIG_THEME_COLOR_CLASSES_COUNT "/ewl/theme/color/classes/count"`
- 8.160.1.12 `#define EWL_CONFIG_THEME_COLOR_CLASSES_OVERRIDE "/ewl/theme/color/classes/override"`
- 8.160.1.13 `#define EWL_CONFIG_THEME_ICON_SIZE "/ewl/theme/icon/size"`
- 8.160.1.14 `#define EWL_CONFIG_THEME_ICON_THEME "/ewl/theme/icon/theme"`
- 8.160.1.15 `#define EWL_CONFIG_THEME_NAME "/ewl/theme/name"`
- 8.160.1.16 `#define EWL_CONFIG_THEME_PRINT_KEYS "/ewl/theme/print/keys"`
- 8.160.1.17 `#define EWL_CONFIG_THEME_PRINT_SIGNALS "/ewl/theme/print/signals"`
- 8.160.1.18 `#define PATH_MAX 4096`

8.160.2 Variable Documentation

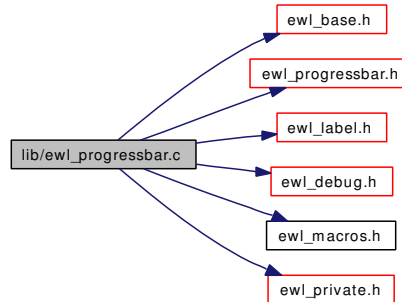
8.160.2.1 `Ewl_Config* ewl_config`

The system configuration data

8.161 lib/ewl_progressbar.c File Reference

```
#include "ewl_base.h"
#include "ewl_progressbar.h"
#include "ewl_label.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_progressbar.c:



Functions

- void **ewl_progressbar_cb_child_resize** (Ewl_Container *c, Ewl_Widget *w, int size __UNUSED__, Ewl_Orientation o __UNUSED__)
- void **ewl_progressbar_cb_child_show** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_progressbar_cb_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_progressbar_cb_value_changed** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_progressbar_custom_label_set** (Ewl_Progressbar *p, char *format_string)

Sets the given format string on the progressbar (lf of lf beers).

- int **ewl_progressbar_init** (Ewl_Progressbar *p)

Initialize the progressbar to some sane starting values.

- void **ewl_progressbar_label_hide** (Ewl_Progressbar *p)

Hides the given progressbars label.

- void **ewl_progressbar_label_set** (Ewl_Progressbar *p, char *label)

Sets the given text on the progressbar.

- void **ewl_progressbar_label_show** (Ewl_Progressbar *p)

Shows the given progressbars label.

- Ewl_Widget * **ewl_progressbar_new** (void)

Allocate and initialize a new progressbar.

8.161.1 Function Documentation

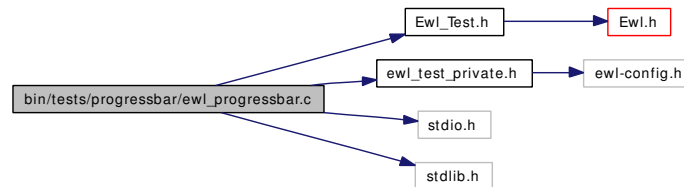
8.161.1.1 void ewl_progressbar_cb_configure (Ewl_Widget * *w*, void *ev_data __ *UNUSED* __, void *user_data __ *UNUSED* __)

8.161.1.2 void ewl_progressbar_cb_value_changed (Ewl_Widget * *w*, void *ev_data __ *UNUSED* __, void *user_data __ *UNUSED* __)

8.162 bin/tests/progressbar/ewl_progressbar.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <stdlib.h>
```

Include dependency graph for ewl_progressbar.c:



Functions

- `void test__info (Ewl_Test *test)`

8.162.1 Function Documentation

8.162.1.1 `void test__info (Ewl_Test * test)`

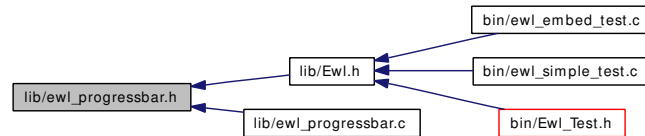
8.163 lib/ewl_progressbar.h File Reference

```
#include "ewl_range.h"
```

Include dependency graph for ewl_progressbar.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Progressbar**

*Inherits from **Ewl_Range** (p. 727) and extends to provide a progressbar widget.*

Defines

- #define **EWL_PROGRESSBAR**(progressbar) ((**Ewl_Progressbar** *) progressbar)
- #define **EWL_PROGRESSBAR_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_PROGRESSBAR_TYPE))
- #define **EWL_PROGRESSBAR_TYPE** "progressbar"

Typedefs

- typedef **Ewl_Progressbar** **Ewl_Progressbar**

Functions

- void **ewl_progressbar_cb_child_resize** (**Ewl_Container** *c, **Ewl_Widget** *w, int size, **Ewl_Orientation** o)
- void **ewl_progressbar_cb_child_show** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_progressbar_cb_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_progressbar_cb_value_changed** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_progressbar_custom_label_set** (**Ewl_Progressbar** *p, char *format_string)

Sets the given format string on the progressbar (lf of lf beers).

- int **ewl_progressbar_init** (**Ewl_Progressbar** *p)

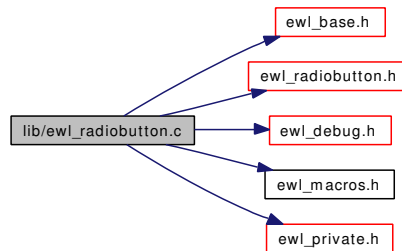
Initialize the progressbar to some sane starting values.

- void **ewl_progressbar_label_hide** (Ewl_Progressbar *p)
Hides the given progressbars label.
- void **ewl_progressbar_label_set** (Ewl_Progressbar *p, char *label)
Sets the given text on the progressbar.
- void **ewl_progressbar_label_show** (Ewl_Progressbar *p)
Shows the given progressbars label.
- **Ewl_Widget ***ewl_progressbar_new (void)
Allocate and initialize a new progressbar.

8.164 lib/ewl_radiobutton.c File Reference

```
#include "ewl_base.h"
#include "ewl_radiobutton.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_radiobutton.c:



Functions

- void **ewl_radiobutton_cb_clicked** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_radiobutton_cb_destroy** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_radiobutton_chain_set** (Ewl_Radiobutton *rb, Ewl_Radiobutton *crb)

Attach the button to a chain of radio buttons.

- int **ewl_radiobutton_init** (Ewl_Radiobutton *rb)

Initialize the radio button fields and callbacks.

- Ewl_Widget * **ewl_radiobutton_new** (void)

Allocate and initialize a new radio button.

8.164.1 Function Documentation

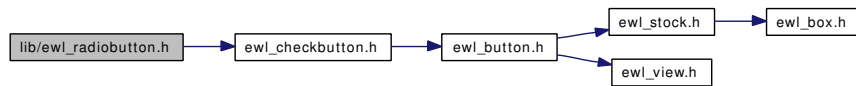
8.164.1.1 void **ewl_radiobutton_cb_clicked** (Ewl_Widget * w, void *ev_data __UNUSED__, void *user_data __UNUSED__)

8.164.1.2 void **ewl_radiobutton_cb_destroy** (Ewl_Widget * w, void *ev_data __UNUSED__, void *user_data __UNUSED__)

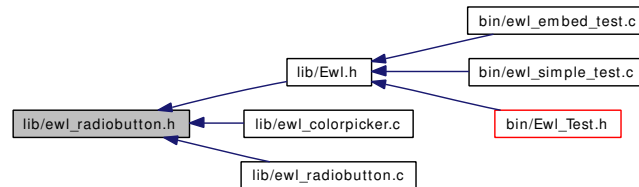
8.165 lib/ewl_radiobutton.h File Reference

```
#include "ewl_checkbutton.h"
```

Include dependency graph for ewl_radiobutton.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Radiobutton**

*Inherits from **Ewl_Checkbutton** (p. 598) and extends it to provide grouping buttons to limit to a single selection in a group at a given time.*

Defines

- #define **EWL_RADIOBUTTON**(button) ((**Ewl_Radiobutton** *) button)
- #define **ewl_radiobutton_checked_set**(r, c) ewl_checkbutton_checked_set(**EWL_CHECKBUTTON**(r), c)
- #define **EWL_RADIOBUTTON_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_RADIOBUTTON_TYPE**))
- #define **ewl_radiobutton_is_checked**(r) ewl_checkbutton_is_checked(**EWL_CHECKBUTTON**(r))
- #define **EWL_RADIOBUTTON_TYPE** "radiobutton"

Typedefs

- typedef **Ewl_Radiobutton** **Ewl_Radiobutton**

Functions

- void **ewl_radiobutton_cb_clicked** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_radiobutton_cb_destroy** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_radiobutton_chain_set** (**Ewl_Radiobutton** *w, **Ewl_Radiobutton** *c)

Attach the button to a chain of radio buttons.

- `int ewl_radiobutton_init (Ewl_Radiobutton *cb)`

Initialize the radio button fields and callbacks.

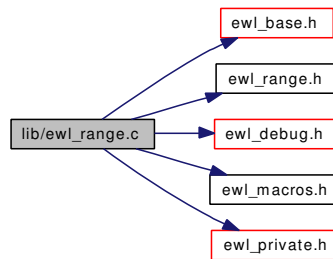
- `Ewl_Widget * ewl_radiobutton_new (void)`

Allocate and initialize a new radio button.

8.166 lib/ewl_range.c File Reference

```
#include "ewl_base.h"
#include "ewl_range.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_range.c:



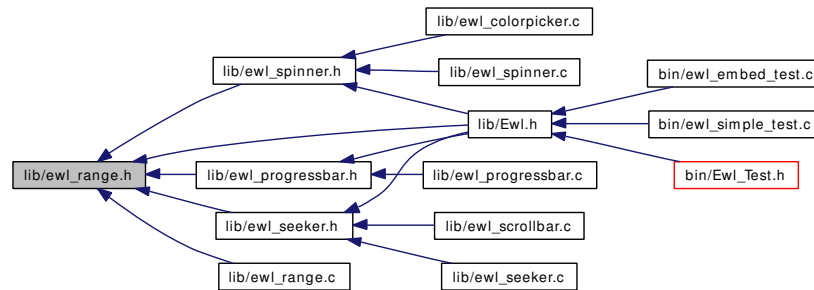
Functions

- void **ewl_range_decrease** (Ewl_Range *r)
Decrease the value of a range by it's step size.
- void **ewl_range_increase** (Ewl_Range *r)
Increase the value of a range by it's step size.
- int **ewl_range_init** (Ewl_Range *r)
Initialize the range to some sane starting values.
- int **ewl_range_invert_get** (Ewl_Range *r)
Retrieve the current invert value from a range.
- void **ewl_range_invert_set** (Ewl_Range *r, int invert)
Changes the invert property on the range for inverting it's scale.
- double **ewl_range_maximum_value_get** (Ewl_Range *r)
Retrieves the maximum value for the range.
- void **ewl_range_maximum_value_set** (Ewl_Range *r, double maxv)
Set the maximum value possible for a range.
- double **ewl_range_minimum_value_get** (Ewl_Range *r)
Retrieves the minimum value for the range.
- void **ewl_range_minimum_value_set** (Ewl_Range *r, double minv)
Set the minimum value possible for a range.

- double **ewl_range_step_get** (**Ewl_Range** *r)
Retrieve the step size of the range.
- void **ewl_range_step_set** (**Ewl_Range** *r, double step)
Set the steps between increments.
- double **ewl_range_value_get** (**Ewl_Range** *r)
Retrieve the current value of the range.
- void **ewl_range_value_set** (**Ewl_Range** *r, double v)
Set the value of pointer of the range.

8.167 lib/ewl_range.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Range**

*Inherits from **Ewl_Container** (p. 618) and extends to provide a basic range widget.*

Defines

- `#define EWL_RANGE(range) ((Ewl_Range *) range)`
- `#define EWL_RANGE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_RANGE_TYPE))`
- `#define EWL_RANGE_TYPE "range"`

Typedefs

- `typedef Ewl_Range Ewl_Range`

Functions

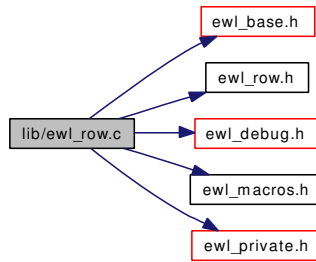
- `void ewl_range_decrease (Ewl_Range *r)`
Decrease the value of a range by it's step size.
- `void ewl_range_increase (Ewl_Range *r)`
Increase the value of a range by it's step size.
- `int ewl_range_init (Ewl_Range *r)`
Initialize the range to some sane starting values.
- `int ewl_range_invert_get (Ewl_Range *r)`
Retrieve the current invert value from a range.
- `void ewl_range_invert_set (Ewl_Range *r, int invert)`
Changes the invert property on the range for inverting it's scale.

- **double ewl_range_maximum_value_get** (Ewl_Range *r)
Retrieves the maximum value for the range.
- **void ewl_range_maximum_value_set** (Ewl_Range *r, double maxv)
Set the maximum value possible for a range.
- **double ewl_range_minimum_value_get** (Ewl_Range *r)
Retrieves the minimum value for the range.
- **void ewl_range_minimum_value_set** (Ewl_Range *r, double minv)
Set the minimum value possible for a range.
- **double ewl_range_step_get** (Ewl_Range *r)
Retrieve the step size of the range.
- **void ewl_range_step_set** (Ewl_Range *r, double step)
Set the steps between increments.
- **double ewl_range_value_get** (Ewl_Range *r)
Retrieve the current value of the range.
- **void ewl_range_value_set** (Ewl_Range *r, double v)
Set the value of pointer of the range.

8.168 lib/ewl_row.c File Reference

```
#include "ewl_base.h"
#include "ewl_row.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_row.c:



Functions

- void **ewl_row_cb_child_hide** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_row_cb_child_resize** (**Ewl_Container** *c, **Ewl_Widget** *w __UNUSED__, int size, **Ewl_Orientation** o)
- void **ewl_row_cb_child_show** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_row_cb_configure** (**Ewl_Widget** *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_row_cb_header_configure** (**Ewl_Widget** *w __UNUSED__, void *ev_data __UNUSED__, void *user_data)
- void **ewl_row_cb_header_destroy** (**Ewl_Widget** *w __UNUSED__, void *ev_data __UNUSED__, void *user_data)
- **Ewl_Widget** * **ewl_row_column_get** (**Ewl_Row** *row, short n)

Retrieve the widget at a specified column.

- void **ewl_row_header_set** (**Ewl_Row** *row, **Ewl_Row** *header)

Set the row header of constraints on cell widths.

- int **ewl_row_init** (**Ewl_Row** *row)

Initialize the row fields of an inheriting object.

- **Ewl_Widget** * **ewl_row_new** (void)

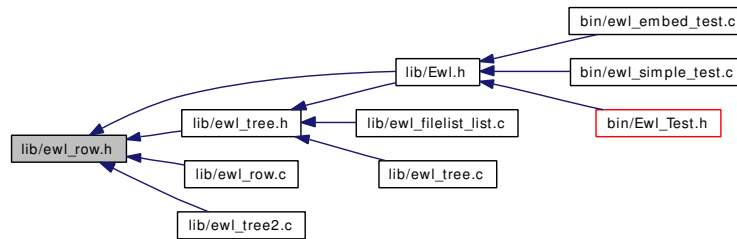
Allocate and initialize a new row.

8.168.1 Function Documentation

- 8.168.1.1 void ewl_row_cb_child_resize (Ewl_Container * *c*, Ewl_Widget *w __ *UNUSED* __, int *size*, Ewl_Orientation *o*)
- 8.168.1.2 void ewl_row_cb_configure (Ewl_Widget * *w*, void *ev_data __ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.168.1.3 void ewl_row_cb_header_configure (Ewl_Widget *w __ *UNUSED* __, void *ev_data __ *UNUSED* __, void * *user_data*)
- 8.168.1.4 void ewl_row_cb_header_destroy (Ewl_Widget *w __ *UNUSED* __, void *ev_data __ *UNUSED* __, void * *user_data*)

8.169 lib/ewl_row.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Row**

*Inherits from **Ewl_Container** (p.618) to create a tree row that lays out cells in a left to right fashion.*

Defines

- `#define EWL_ROW(t) ((Ewl_Row *)t)`
- `#define EWL_ROW_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_ROW_TYPE))`
- `#define EWL_ROW_TYPE "row"`

Typedefs

- `typedef Ewl_Row Ewl_Row`

Functions

- `void ewl_row_cb_child_hide (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_row_cb_child_resize (Ewl_Container *c, Ewl_Widget *w, int size, Ewl_Orientation o)`
- `void ewl_row_cb_child_show (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_row_cb_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_row_cb_header_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_row_cb_header_destroy (Ewl_Widget *w, void *ev_data, void *user_data)`
- `Ewl_Widget * ewl_row_column_get (Ewl_Row *row, short n)`
Retrieve the widget at a specified column.
- `void ewl_row_header_set (Ewl_Row *row, Ewl_Row *header)`
Set the row header of constraints on cell widths.
- `int ewl_row_init (Ewl_Row *row)`

Initialize the row fields of an inheriting object.

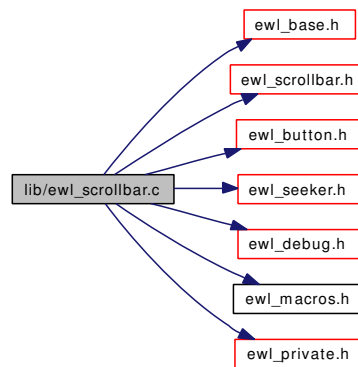
- **Ewl_Widget * ewl_row_new** (void)

Allocate and initialize a new row.

8.170 lib/ewl_scrollbar.c File Reference

```
#include "ewl_base.h"
#include "ewl_scrollbar.h"
#include "ewl_button.h"
#include "ewl_seeker.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_scrollbar.c:



Functions

- **Ewl_Widget * ewl_hscrollbar_new** (void)
Allocate and initialize a new horizontal scrollbar widget.
- **void ewl_scrollbar_cb_scroll_start** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data)
- **void ewl_scrollbar_cb_scroll_stop** (Ewl_Widget *w __UNUSED__, void *ev_data __UNUSED__, void *user_data)
- **int ewl_scrollbar_init** (Ewl_Scrollbar *s)
Initialize a scrollbar to default values.
- **char ewl_scrollbar_inverse_scroll_get** (Ewl_Scrollbar *s)
Checks the inverse scrolling flag on a scrollbar.
- **void ewl_scrollbar_inverse_scroll_set** (Ewl_Scrollbar *s, char i)
Sets the inverse scrolling flag on a scrollbar.
- **Ewl_Widget * ewl_scrollbar_new** (void)
Allocate and initialize a new scrollbar widget.
- **Ewl_Orientation ewl_scrollbar_orientation_get** (Ewl_Scrollbar *s)
Change the orientation of a scrollbar.

- void **ewl_scrollbar_orientation_set** (Ewl_Scrollbar *s, Ewl_Orientation o)
Change the orientation of a scrollbar.
- double **ewl_scrollbar_step_get** (Ewl_Scrollbar *s)
Get the current step size of the scrollbar.
- void **ewl_scrollbar_step_set** (Ewl_Scrollbar *s, double v)
Change the step size of a scrollbar.
- double **ewl_scrollbar_value_get** (Ewl_Scrollbar *s)
Get the current value of the dragbar.
- void **ewl_scrollbar_value_set** (Ewl_Scrollbar *s, double v)
Set the current value of the dragbar.
- Ewl_Widget * **ewl_vscrollbar_new** (void)
Allocate and initialize a new vertical scrollbar widget.

8.170.1 Function Documentation

- 8.170.1.1 void **ewl_scrollbar_cb_scroll_start** (Ewl_Widget * w, void *ev_data __ *UNUSED* __, void * *user_data*)
- 8.170.1.2 void **ewl_scrollbar_cb_scroll_stop** (Ewl_Widget *w __ *UNUSED* __, void *ev_data __ *UNUSED* __, void * *user_data*)

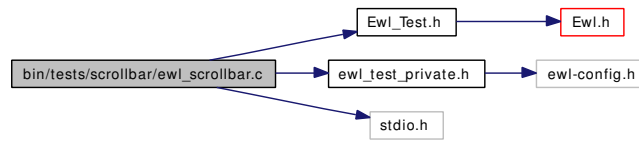
8.171 bin/tests/scrollbar/ewl_scrollbar.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_scrollbar.c:



Functions

- `void test_info (Ewl_Test *test)`

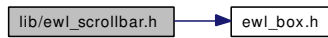
8.171.1 Function Documentation

8.171.1.1 `void test_info (Ewl_Test * test)`

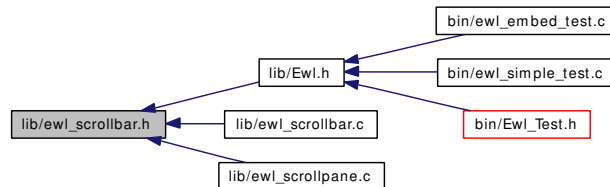
8.172 lib/ewl_scrollbar.h File Reference

```
#include "ewl_box.h"
```

Include dependency graph for ewl_scrollbar.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Scrollbar**

*Inherits from **Ewl_Box** (p. 586) and extends to provide a scrollbar widget.*

Defines

- #define **EWL_SCROLLBAR**(scrollbar) ((**Ewl_Scrollbar** *) scrollbar)
- #define **EWL_SCROLLBAR_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SCROLLBAR_TYPE))
- #define **EWL_SCROLLBAR_TYPE** "scrollbar"

Typedefs

- typedef **Ewl_Scrollbar** **Ewl_Scrollbar**

Functions

- **Ewl_Widget** * **ewl_hscrollbar_new** (void)
Allocate and initialize a new horizontal scrollbar widget.
- void **ewl_scrollbar_cb_scroll_start** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_scrollbar_cb_scroll_stop** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- int **ewl_scrollbar_init** (**Ewl_Scrollbar** *s)
Initialize a scrollbar to default values.
- char **ewl_scrollbar_inverse_scroll_get** (**Ewl_Scrollbar** *s)

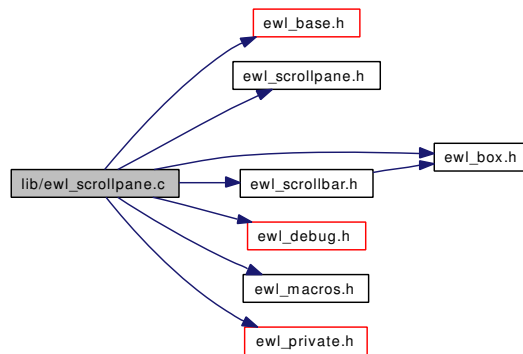
Checks the inverse scrolling flag on a scrollbar.

- **void ewl_scrollbar_inverse_scroll_set** (**Ewl_Scrollbar** *s, char v)
Sets the inverse scrolling flag on a scrollbar.
- **Ewl_Widget * ewl_scrollbar_new** (void)
Allocate and initialize a new scrollbar widget.
- **Ewl_Orientation ewl_scrollbar_orientation_get** (**Ewl_Scrollbar** *s)
Change the orientation of a scrollbar.
- **void ewl_scrollbar_orientation_set** (**Ewl_Scrollbar** *s, **Ewl_Orientation** orientation)
Change the orientation of a scrollbar.
- **double ewl_scrollbar_step_get** (**Ewl_Scrollbar** *s)
Get the current step size of the scrollbar.
- **void ewl_scrollbar_step_set** (**Ewl_Scrollbar** *s, double v)
Change the step size of a scrollbar.
- **double ewl_scrollbar_value_get** (**Ewl_Scrollbar** *s)
Get the current value of the dragbar.
- **void ewl_scrollbar_value_set** (**Ewl_Scrollbar** *s, double v)
Set the current value of the dragbar.
- **Ewl_Widget * ewl_vscrollbar_new** (void)
Allocate and initialize a new vertical scrollbar widget.

8.173 lib/ewl_scrollpane.c File Reference

```
#include "ewl_base.h"
#include "ewl_scrollpane.h"
#include "ewl_box.h"
#include "ewl_scrollbar.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_scrollpane.c:



Functions

- void **ewl_scrollpane_cb_child_resize** (Ewl_Container *parent, Ewl_Widget *child)
- void **ewl_scrollpane_cb_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_scrollpane_cb_focus_jump** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_scrollpane_cb_hscroll** (Ewl_Widget *w __UNUSED__, void *ev_data __UNUSED__, void *user_data)
- void **ewl_scrollpane_cb_vscroll** (Ewl_Widget *w __UNUSED__, void *ev_data __UNUSED__, void *user_data)
- void **ewl_scrollpane_cb_wheel_scroll** (Ewl_Widget *cb, void *ev_data, void *user_data __UNUSED__)
- Ewl_Scrollpane_Flags **ewl_scrollpane_hscrollbar_flag_get** (Ewl_Scrollpane *s)

Get flags for horizontal scrollbar.

- void **ewl_scrollpane_hscrollbar_flag_set** (Ewl_Scrollpane *s, Ewl_Scrollpane_Flags f)

Set flags for horizontal scrollbar.

- double **ewl_scrollpane_hscrollbar_step_get** (Ewl_Scrollpane *s)

Retrives the value of the stepping of the vertical scrollbar in s.

- **double ewl_scrollpane_hscrollbar_value_get** (Ewl_Scrollpane *s)
Retrieves the value of the horizontal scrollbar in s.
- **void ewl_scrollpane_hscrollbar_value_set** (Ewl_Scrollpane *s, double val)
Set the value of the horizontal scrollbar in s to val.
- **int ewl_scrollpane_init** (Ewl_Scrollpane *s)
Initialize the fields of a scrollpane.
- **Ewl_Widget * ewl_scrollpane_new** (void)
Create a new scrollpane.
- **Ewl_Scrollpane_Flags ewl_scrollpane_vscrollbar_flag_get** (Ewl_Scrollpane *s)
Get flags for vertical scrollbar.
- **void ewl_scrollpane_vscrollbar_flag_set** (Ewl_Scrollpane *s, Ewl_Scrollpane_Flags f)
Set flags for vertical scrollbar.
- **double ewl_scrollpane_vscrollbar_step_get** (Ewl_Scrollpane *s)
Retrieves the value of the stepping of the vertical scrollbar in s.
- **double ewl_scrollpane_vscrollbar_value_get** (Ewl_Scrollpane *s)
Retrieves the value of the vertical scrollbar in s.
- **void ewl_scrollpane_vscrollbar_value_set** (Ewl_Scrollpane *s, double val)
Set the value of the vertical scrollbar in s to val.

8.173.1 Function Documentation

- 8.173.1.1 void ewl_scrollpane_cb_configure** (Ewl_Widget * w, void *ev_data __ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.173.1.2 void ewl_scrollpane_cb_focus_jump** (Ewl_Widget * w, void *ev_data __ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.173.1.3 void ewl_scrollpane_cb_hscroll** (Ewl_Widget *w __ *UNUSED* __, void *ev_data __ *UNUSED* __, void * user_data)
- 8.173.1.4 void ewl_scrollpane_cb_vscroll** (Ewl_Widget *w __ *UNUSED* __, void *ev_data __ *UNUSED* __, void * user_data)
- 8.173.1.5 void ewl_scrollpane_cb_wheel_scroll** (Ewl_Widget * cb, void * ev_data, void *user_data __ *UNUSED* __)

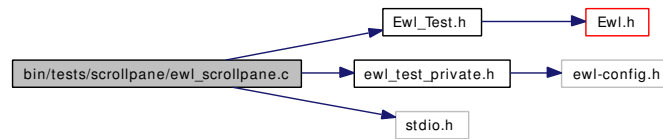
8.174 bin/tests/scrollpane/ewl_scrollpane.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_scrollpane.c:



Functions

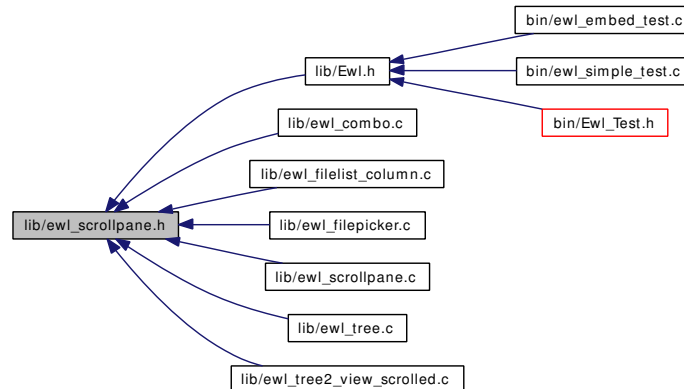
- `void test_info (Ewl_Test *test)`

8.174.1 Function Documentation

8.174.1.1 `void test_info (Ewl_Test * test)`

8.175 lib/ewl_scrollpane.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Scrollpane**

*Inherits from **Ewl_Container** (p. 618) and extends it to enclose a single widget in vertical and horizontal scrollbars for panning around displaying different regions of the enclosed widget.*

Defines

- `#define EWL_SCROLLPANE(scrollpane) ((Ewl_Scrollpane *) scrollpane)`
- `#define EWL_SCROLLPANE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SCROLLPANE_TYPE))`
- `#define EWL_SCROLLPANE_TYPE "scrollpane"`

Typedefs

- `typedef Ewl_Scrollpane Ewl_Scrollpane`

Functions

- `void ewl_scrollpane_cb_child_resize (Ewl_Container *parent, Ewl_Widget *child)`
- `void ewl_scrollpane_cb_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_scrollpane_cb_focus_jump (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_scrollpane_cb_hscroll (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_scrollpane_cb_vscroll (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_scrollpane_cb_wheel_scroll (Ewl_Widget *w, void *ev_data, void *user_data)`
- `Ewl_Scrollpane_Flags ewl_scrollpane_hscrollbar_flag_get (Ewl_Scrollpane *s)`

Get flags for horizontal scrollbar.

- void ewl_scrollpane_hscrollbar_flag_set (Ewl_Scrollpane *s, Ewl_Scrollpane_Flags f)

Set flags for horizontal scrollbar.

- double ewl_scrollpane_hscrollbar_step_get (Ewl_Scrollpane *s)

Retrives the value of the stepping of the vertical scrollbar in s.

- double ewl_scrollpane_hscrollbar_value_get (Ewl_Scrollpane *s)

Retrieves the value of the horizontal scrollbar in s.

- void ewl_scrollpane_hscrollbar_value_set (Ewl_Scrollpane *s, double val)

Set the value of the horizontal scrollbar in s to val.

- int ewl_scrollpane_init (Ewl_Scrollpane *s)

Initialize the fields of a scrollpane.

- Ewl_Widget * ewl_scrollpane_new (void)

Create a new scrollpane.

- Ewl_Scrollpane_Flags ewl_scrollpane_vscrollbar_flag_get (Ewl_Scrollpane *s)

Get flags for vertical scrollbar.

- void ewl_scrollpane_vscrollbar_flag_set (Ewl_Scrollpane *s, Ewl_Scrollpane_Flags f)

Set flags for vertical scrollbar.

- double ewl_scrollpane_vscrollbar_step_get (Ewl_Scrollpane *s)

Retrives the value of the stepping of the vertical scrollbar in s.

- double ewl_scrollpane_vscrollbar_value_get (Ewl_Scrollpane *s)

Retrieves the value of the vertical scrollbar in s.

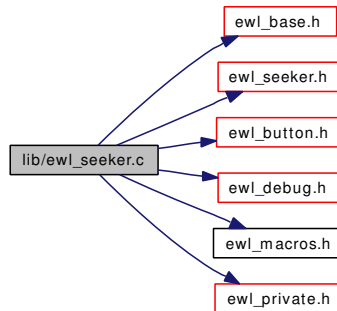
- void ewl_scrollpane_vscrollbar_value_set (Ewl_Scrollpane *s, double val)

Set the value of the vertical scrollbar in s to val.

8.176 lib/ewl_seeker.c File Reference

```
#include "ewl_base.h"
#include "ewl_seeker.h"
#include "ewl_button.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_seeker.c:



Functions

- **Ewl_Widget * ewl_hseeker_new** (void)
Allocate and initialize a new seeker with horizontal orientation.
- **int ewl_seeker_autohide_get** (Ewl_Seeker *s)
Retrieves the current autohide setting on a seeker.
- **void ewl_seeker_autohide_set** (Ewl_Seeker *s, int v)
Changes the autohide setting on the seeker to v.
- **void ewl_seeker_cb_button_mouse_down** (Ewl_Widget *w, void *ev_data, void *user_data __UNUSED__)
- **void ewl_seeker_cb_button_mouse_up** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- **void ewl_seeker_cb_child_show** (Ewl_Container *p, Ewl_Widget *w)
- **void ewl_seeker_cb_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- **void ewl_seeker_cb_key_down** (Ewl_Widget *w, void *ev_data, void *user_data __UNUSED__)
- **void ewl_seeker_cb_mouse_down** (Ewl_Widget *w, void *ev_data, void *user_data __UNUSED__)
- **void ewl_seeker_cb_mouse_move** (Ewl_Widget *w, void *ev_data, void *user_data __UNUSED__)
- **void ewl_seeker_cb_mouse_up** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- **int ewl_seeker_init** (Ewl_Seeker *s)

Initialize the seeker to some sane starting values.

- **Ewl_Widget * ewl_seeker_new (void)**
Allocate and initialize a new seeker with default orientation.
- **Ewl_Orientation ewl_seeker_orientation_get (Ewl_Seeker *s)**
This will retrieve the current orientation set on the seeker.
- **void ewl_seeker_orientation_set (Ewl_Seeker *s, Ewl_Orientation o)**
Changes the orientation of the given seeker.
- **Ewl_Widget * ewl_vseeker_new (void)**
Allocate and initialize a new seeker with vertical orientation.

8.176.1 Function Documentation

- 8.176.1.1 void ewl_seeker_cb_button_mouse_down (Ewl_Widget * w, void * ev_data, void *user_data __UNUSED__)**
- 8.176.1.2 void ewl_seeker_cb_button_mouse_up (Ewl_Widget * w, void *ev_data __UNUSED__, void *user_data __UNUSED__)**
- 8.176.1.3 void ewl_seeker_cb_configure (Ewl_Widget * w, void *ev_data __UNUSED__, void *user_data __UNUSED__)**
- 8.176.1.4 void ewl_seeker_cb_key_down (Ewl_Widget * w, void * ev_data, void *user_data __UNUSED__)**
- 8.176.1.5 void ewl_seeker_cb_mouse_down (Ewl_Widget * w, void * ev_data, void *user_data __UNUSED__)**
- 8.176.1.6 void ewl_seeker_cb_mouse_move (Ewl_Widget * w, void * ev_data, void *user_data __UNUSED__)**
- 8.176.1.7 void ewl_seeker_cb_mouse_up (Ewl_Widget * w, void *ev_data __UNUSED__, void *user_data __UNUSED__)**

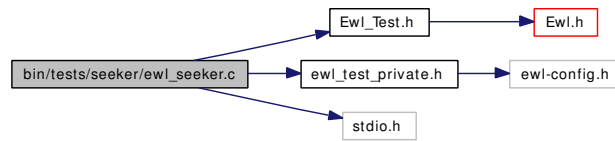
8.177 bin/tests/seeker/ewl_seeker.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_seeker.c:



Functions

- `void test_info (Ewl_Test *test)`

8.177.1 Function Documentation

8.177.1.1 `void test_info (Ewl_Test * test)`

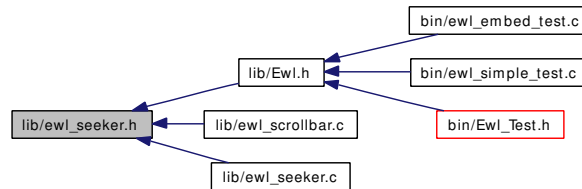
8.178 lib/ewl_seeker.h File Reference

```
#include "ewl_range.h"
```

Include dependency graph for ewl_seeker.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Seeker**

*Inherits from **Ewl_Range** (p. 727) and extends to provide a seeker widget.*

Defines

- `#define EWL_SEEKER(seeker) ((Ewl_Seeker *) seeker)`
- `#define EWL_SEEKER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SEEKER_TYPE))`
- `#define EWL_SEEKER_TYPE "seeker"`

Typedefs

- `typedef Ewl_Seeker Ewl_Seeker`

Functions

- `Ewl_Widget * ewl_hseeker_new (void)`
Allocate and initialize a new seeker with horizontal orientation.
- `int ewl_seeker_autohide_get (Ewl_Seeker *s)`
Retrieves the current autohide setting on a seeker.
- `void ewl_seeker_autohide_set (Ewl_Seeker *s, int v)`
Changes the autohide setting on the seeker to v.
- `void ewl_seeker_cb_button_mouse_down (Ewl_Widget *w, void *ev_data, void *user_data)`

- void **ewl_seeker_cb_button_mouse_up** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_seeker_cb_child_show** (**Ewl_Container** *p, **Ewl_Widget** *w)
- void **ewl_seeker_cb_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_seeker_cb_key_down** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_seeker_cb_mouse_down** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_seeker_cb_mouse_move** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_seeker_cb_mouse_up** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- int **ewl_seeker_init** (**Ewl_Seeker** *s)

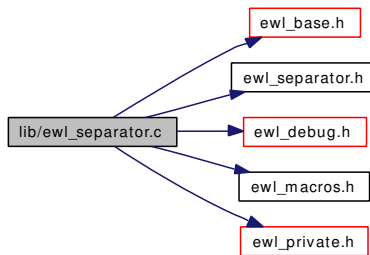
Initialize the seeker to some sane starting values.

- **Ewl_Widget** * **ewl_seeker_new** (void)
Allocate and initialize a new seeker with default orientation.
- **Ewl_Orientation** **ewl_seeker_orientation_get** (**Ewl_Seeker** *s)
This will retrieve the current orientation set on the seeker.
- void **ewl_seeker_orientation_set** (**Ewl_Seeker** *s, **Ewl_Orientation** o)
Changes the orientation of the given seeker.
- **Ewl_Widget** * **ewl_vseeker_new** (void)
Allocate and initialize a new seeker with vertical orientation.

8.179 lib/ewl_separator.c File Reference

```
#include "ewl_base.h"
#include "ewl_separator.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_separator.c:

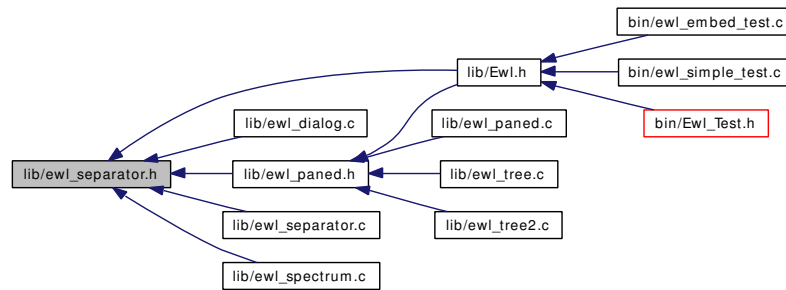


Functions

- **Ewl_Widget * ewl_hseparator_new (void)**
Allocate a new separator widget with horizontal orientation.
- **int ewl_separator_init (Ewl_Separator *s)**
Initialize the separator and inherited fields.
- **Ewl_Widget * ewl_separator_new (void)**
Allocate a new separator widget with default (horizontal) orientation.
- **Ewl_Orientation ewl_separator_orientation_get (Ewl_Separator *s)**
This retrieves the current orientation of the separator.
- **void ewl_separator_orientation_set (Ewl_Separator *s, Ewl_Orientation o)**
Set the orientation of the separator.
- **Ewl_Widget * ewl_vseparator_new (void)**
Allocate a new separator widget with vertical orientation.

8.180 lib/ewl_separator.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Separator**

*Inherits from **Ewl_Widget** (p. 785) to allow drawing a horizontal or vertical line between widgets, depending on the orientation the **Ewl_Separator** (p. 740).*

Defines

- `#define EWL_SEPARATOR(separator) ((Ewl_Separator *) separator)`
- `#define EWL_SEPARATOR_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SEPARATOR_TYPE))`
- `#define EWL_SEPARATOR_TYPE "separator"`

Typedefs

- `typedef Ewl_Separator Ewl_Separator`

Functions

- **Ewl_Widget * ewl_hseparator_new** (void)
Allocate a new separator widget with horizontal orientation.
- **int ewl_separator_init** (Ewl_Separator *s)
Initialize the separator and inherited fields.
- **Ewl_Widget * ewl_separator_new** (void)
Allocate a new separator widget with default (horizontal) orientation.
- **Ewl_Orientation ewl_separator_orientation_get** (Ewl_Separator *s)
This retrieves the current orientation of the separator.
- **void ewl_separator_orientation_set** (Ewl_Separator *s, Ewl_Orientation o)
Set the orientation of the separator.

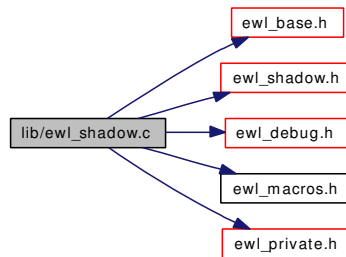
- **Ewl_Widget * ewl_vseparator_new** (void)

Allocate a new separator widget with vertical orientation.

8.181 lib/ewl_shadow.c File Reference

```
#include "ewl_base.h"
#include "ewl_shadow.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_shadow.c:



Functions

- **int ewl_shadow_init (Ewl_Shadow *s)**
Initialize a shadow container to default values.
- **Ewl_Widget * ewl_shadow_new (void)**
Allocate and initialize a new shadow container.

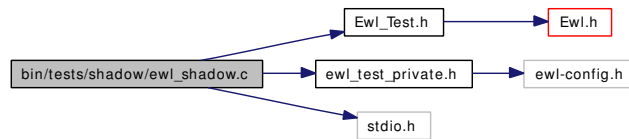
8.182 bin/tests/shadow/ewl_shadow.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_shadow.c:



Functions

- `void test_info (Ewl_Test *test)`

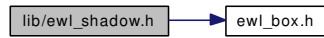
8.182.1 Function Documentation

8.182.1.1 `void test_info (Ewl_Test * test)`

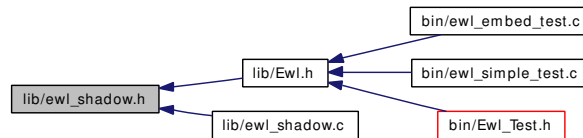
8.183 lib/ewl_shadow.h File Reference

```
#include "ewl_box.h"
```

Include dependency graph for ewl_shadow.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Shadow**

*Inherits from **Ewl_Box** (p. 586) to allow drawing a border and shadow decoration around widgets.*

Defines

- `#define EWL_SHADOW(shadow) ((Ewl_Shadow *) shadow)`
- `#define EWL_SHADOW_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SHADOW_TYPE))`
- `#define EWL_SHADOW_TYPE "shadow"`

Typedefs

- `typedef Ewl_Shadow Ewl_Shadow`

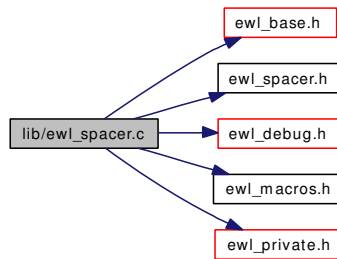
Functions

- `int ewl_shadow_init (Ewl_Shadow *s)`
Initialize a shadow container to default values.
- `Ewl_Widget * ewl_shadow_new (void)`
Allocate and initialize a new shadow container.

8.184 lib/ewl_spacer.c File Reference

```
#include "ewl_base.h"
#include "ewl_spacer.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_spacer.c:

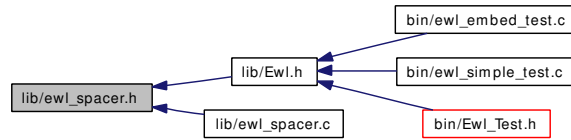


Functions

- `int ewl_spacer_init (Ewl_Spacer *s)`
Initialize a spacer to starting values.
- `Ewl_Widget * ewl_spacer_new (void)`
Allocate and initialize a new spacer.

8.185 lib/ewl_spacer.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Spacer**

*Inherits from **Ewl_Widget** (p. 785) and provides as a spacer between widgets.*

Defines

- `#define EWL_SPACER(spacer) ((Ewl_Spacer *) spacer)`
- `#define EWL_SPACER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SPACER_TYPE))`
- `#define EWL_SPACER_TYPE "spacer"`

Typedefs

- `typedef Ewl_Spacer Ewl_Spacer`

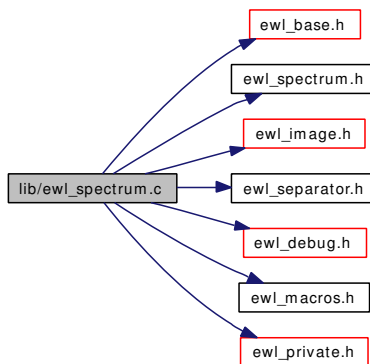
Functions

- `int ewl_spacer_init (Ewl_Spacer *s)`
Initialize a spacer to starting values.
- `Ewl_Widget * ewl_spacer_new (void)`
Allocate and initialize a new spacer.

8.186 lib/ewl_spectrum.c File Reference

```
#include "ewl_base.h"
#include "ewl_spectrum.h"
#include "ewl_image.h"
#include "ewl_separator.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_spectrum.c:



Functions

- void **ewl_spectrum_canvas_cb_reveal** (Ewl_Widget *w __UNUSED__, void *ev __UNUSED__, void *data)
- void **ewl_spectrum_cb_configure** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_spectrum_cb_mouse_down** (Ewl_Widget *w, void *ev, void *data __UNUSED__)
- void **ewl_spectrum_cb_mouse_move** (Ewl_Widget *w, void *ev, void *data __UNUSED__)
- void **ewl_spectrum_cb_mouse_up** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_spectrum_hsv_get** (Ewl_Spectrum *sp, double *h, double *s, double *v)
Get the HSV value from the spectrum.
- void **ewl_spectrum_hsv_set** (Ewl_Spectrum *sp, double h, double s, double v)
Set the HSV values for the spectrum.
- int **ewl_spectrum_init** (Ewl_Spectrum *sp)
Initializes an Ewl_Spectrum widget to default values.
- Ewl_Color_Mode **ewl_spectrum_mode_get** (Ewl_Spectrum *sp)
Get the mode of the spectrum.

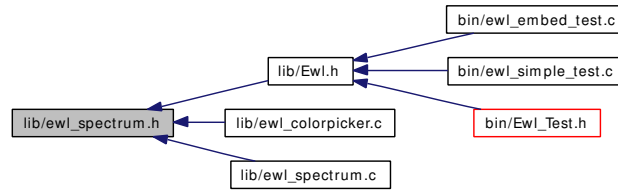
- void **ewl_spectrum_mode_set** (**Ewl_Spectrum** *sp, **Ewl_Color_Mode** mode)
Set the mode of the spectrum.
- **Ewl_Widget** * **ewl_spectrum_new** (void)
*Creates a new **Ewl_Spectrum** (p. 743) widget.*
- void **ewl_spectrum_rgb_get** (**Ewl_Spectrum** *sp, unsigned int *r, unsigned int *g, unsigned int *b)
Get the RGB values for the spectrum.
- void **ewl_spectrum_rgb_set** (**Ewl_Spectrum** *sp, unsigned int r, unsigned int g, unsigned int b)
Set the RGB values for the spectrum.
- **Ewl_Spectrum_Type** **ewl_spectrum_type_get** (**Ewl_Spectrum** *sp)
Get the type of the spectrum.
- void **ewl_spectrum_type_set** (**Ewl_Spectrum** *sp, **Ewl_Spectrum_Type** type)
Set the type of the spectrum.

8.186.1 Function Documentation

- 8.186.1.1 void **ewl_spectrum_canvas_cb_reveal** (**Ewl_Widget** *w
__ **UNUSED** __, void *ev __ **UNUSED** __, void * data)
- 8.186.1.2 void **ewl_spectrum_cb_configure** (**Ewl_Widget** * w, void *ev
__ **UNUSED** __, void *data __ **UNUSED** __)
- 8.186.1.3 void **ewl_spectrum_cb_mouse_down** (**Ewl_Widget** * w, void * ev, void
*data __ **UNUSED** __)
- 8.186.1.4 void **ewl_spectrum_cb_mouse_move** (**Ewl_Widget** * w, void * ev, void
*data __ **UNUSED** __)
- 8.186.1.5 void **ewl_spectrum_cb_mouse_up** (**Ewl_Widget** * w, void *ev
__ **UNUSED** __, void *data __ **UNUSED** __)

8.187 lib/ewl_spectrum.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Spectrum**

*Inherits from **Ewl_Overlay** (p. 712) and extends to provide a colour spectrum.*

Defines

- #define **EWL_SPECTRUM**(sp) ((**Ewl_Spectrum** *)sp)
- #define **EWL_SPECTRUM_IS**(w) (ewl_widget_type_is(**EWL_WIDGET**(w), **EWL_SPECTRUM_TYPE**))
- #define **EWL_SPECTRUM_TYPE** "spectrum"

Typedefs

- typedef **Ewl_Spectrum** **Ewl_Spectrum**

Functions

- void **ewl_spectrum_canvas_cb_reveal** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_spectrum_cb_configure** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_spectrum_cb_mouse_down** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_spectrum_cb_mouse_move** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_spectrum_cb_mouse_up** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_spectrum_hsv_get** (**Ewl_Spectrum** *sp, double *h, double *s, double *v)

Get the HSV value from the spectrum.

- void **ewl_spectrum_hsv_set** (**Ewl_Spectrum** *sp, double h, double s, double v)

Set the HSV values for the spectrum.

- int **ewl_spectrum_init** (**Ewl_Spectrum** *sp)

Initializes an Ewl_Spectrum widget to default values.

- **Ewl_Color_Mode** **ewl_spectrum_mode_get** (**Ewl_Spectrum** *sp)

Get the mode of the spectrum.

- void **ewl_spectrum_mode_set** (**Ewl_Spectrum** *sp, **Ewl_Color_Mode** mode)

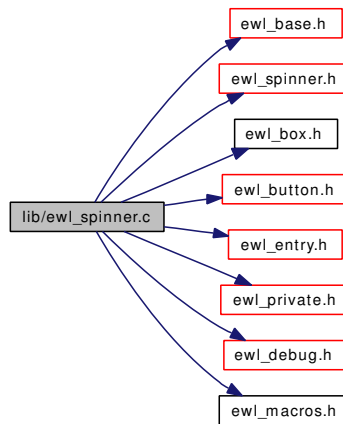
Set the mode of the spectrum.

- **Ewl_Widget * ewl_spectrum_new** (void)
*Creates a new **Ewl_Spectrum** (p. 743) widget.*
- **void ewl_spectrum_rgb_get** (**Ewl_Spectrum** *sp, unsigned int *r, unsigned int *g, unsigned int *b)
Get the RGB values for the spectrum.
- **void ewl_spectrum_rgb_set** (**Ewl_Spectrum** *sp, unsigned int r, unsigned int g, unsigned int b)
Set the RGB values for the spectrum.
- **Ewl_Spectrum_Type ewl_spectrum_type_get** (**Ewl_Spectrum** *sp)
Get the type of the spectrum.
- **void ewl_spectrum_type_set** (**Ewl_Spectrum** *sp, **Ewl_Spectrum_Type** type)
Set the type of the spectrum.

8.188 lib/ewl_spinner.c File Reference

```
#include "ewl_base.h"
#include "ewl_spinner.h"
#include "ewl_box.h"
#include "ewl_button.h"
#include "ewl_entry.h"
#include "ewl_private.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
```

Include dependency graph for ewl_spinner.c:



Functions

- void **ewl_spinner_cb_child_resize** (Ewl_Container *c, Ewl_Widget *w, int size __UNUSED__, Ewl_Orientation o __UNUSED__)
- void **ewl_spinner_cb_child_show** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_spinner_cb_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_spinner_cb_decrease_value** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_spinner_cb_destroy** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_spinner_cb_focus_out** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_spinner_cb_increase_value** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_spinner_cb_key_down** (Ewl_Widget *w, void *ev_data, void *user_data __UNUSED__)
- void **ewl_spinner_cb_realize** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_spinner_cb_value_changed** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)

- void `ewl_spinner_cb_value_stop` (`Ewl_Widget *w` `__UNUSED__`, void *`ev_data` `__UNUSED__`, void *`user_data`)
- void `ewl_spinner_cb_wheel` (`Ewl_Widget *w`, void *`ev_data`, void *`user_data` `__UNUSED__`)
- unsigned char `ewl_spinner_digits_get` (`Ewl_Spinner *s`)
This will retrieve the number of digits displayed by the spinner.
- void `ewl_spinner_digits_set` (`Ewl_Spinner *s`, unsigned char `digits`)
Set the number of digits displayed by a spinner.
- int `ewl_spinner_init` (`Ewl_Spinner *s`)
Initialize a spinner to default values and callbacks.
- `Ewl_Widget *` `ewl_spinner_new` (void)
Allocate a new spinner widget.

8.188.1 Function Documentation

- 8.188.1.1 void `ewl_spinner_cb_configure` (`Ewl_Widget * w`, void *`ev_data` `__UNUSED__`, void *`user_data` `__UNUSED__`)
- 8.188.1.2 void `ewl_spinner_cb_destroy` (`Ewl_Widget * w`, void *`ev_data` `__UNUSED__`, void *`user_data` `__UNUSED__`)
- 8.188.1.3 void `ewl_spinner_cb_focus_out` (`Ewl_Widget * w`, void *`ev_data` `__UNUSED__`, void *`user_data` `__UNUSED__`)
- 8.188.1.4 void `ewl_spinner_cb_key_down` (`Ewl_Widget * w`, void * `ev_data`, void *`user_data` `__UNUSED__`)
- 8.188.1.5 void `ewl_spinner_cb_realize` (`Ewl_Widget * w`, void *`ev_data` `__UNUSED__`, void *`user_data` `__UNUSED__`)
- 8.188.1.6 void `ewl_spinner_cb_value_changed` (`Ewl_Widget * w`, void *`ev_data` `__UNUSED__`, void *`user_data` `__UNUSED__`)
- 8.188.1.7 void `ewl_spinner_cb_value_stop` (`Ewl_Widget *w` `__UNUSED__`, void *`ev_data` `__UNUSED__`, void * `user_data`)
- 8.188.1.8 void `ewl_spinner_cb_wheel` (`Ewl_Widget * w`, void * `ev_data`, void *`user_data` `__UNUSED__`)

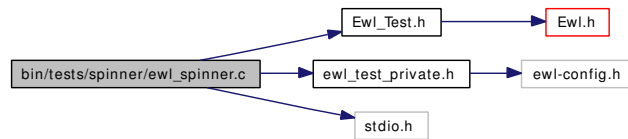
8.189 bin/tests/spinner/ewl_spinner.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_spinner.c:



Functions

- `void test_info (Ewl_Test *test)`

8.189.1 Function Documentation

8.189.1.1 `void test_info (Ewl_Test * test)`

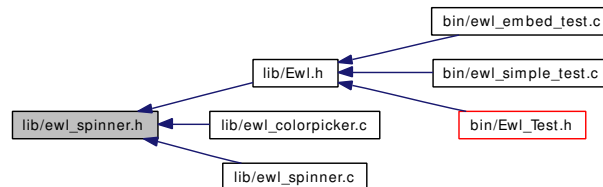
8.190 lib/ewl_spinner.h File Reference

```
#include "ewl_range.h"
```

Include dependency graph for ewl_spinner.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Spinner**

*Inherits from **Ewl_Range** (p. 727) and extends to provide a spinner widget.*

Defines

- `#define EWL_SPINNER(spinner) ((Ewl_Spinner *) spinner)`
- `#define EWL_SPINNER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_SPINNER_TYPE))`
- `#define EWL_SPINNER_TYPE "spinner"`

Typedefs

- `typedef Ewl_Spinner Ewl_Spinner`

Functions

- `void ewl_spinner_cb_child_resize (Ewl_Container *c, Ewl_Widget *w, int size, Ewl_Orientation o)`
- `void ewl_spinner_cb_child_show (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_spinner_cb_configure (Ewl_Widget *widget, void *ev_data, void *user_data)`
- `void ewl_spinner_cb_decrease_value (Ewl_Widget *widget, void *ev_data, void *user_data)`
- `void ewl_spinner_cb_destroy (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_spinner_cb_focus_out (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_spinner_cb_increase_value (Ewl_Widget *widget, void *ev_data, void *user_data)`

- void **ewl_spinner_cb_key_down** (Ewl_Widget *widget, void *ev_data, void *user_data)
- void **ewl_spinner_cb_realize** (Ewl_Widget *widget, void *ev_data, void *user_data)
- void **ewl_spinner_cb_value_changed** (Ewl_Widget *widget, void *ev_data, void *user_data)
- void **ewl_spinner_cb_value_stop** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_spinner_cb_wheel** (Ewl_Widget *w, void *ev_data, void *user_data)
- unsigned char **ewl_spinner_digits_get** (Ewl_Spinner *s)

This will retrieve the number of digits displayed by the spinner.

- void **ewl_spinner_digits_set** (Ewl_Spinner *s, unsigned char digits)

Set the number of digits displayed by a spinner.

- int **ewl_spinner_init** (Ewl_Spinner *s)

Initialize a spinner to default values and callbacks.

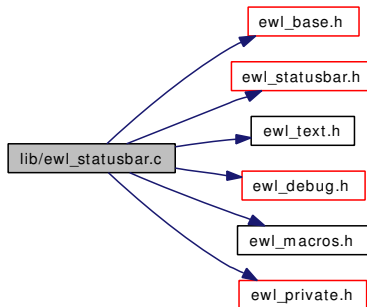
- Ewl_Widget * **ewl_spinner_new** (void)

Allocate a new spinner widget.

8.191 lib/ewl_statusbar.c File Reference

```
#include "ewl_base.h"
#include "ewl_statusbar.h"
#include "ewl_text.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_statusbar.c:



Functions

- void **ewl_statusbar_active_set** (Ewl_Statusbar *sb, Ewl_Position pos)
Sets the active statusbar container.
- void **ewl_statusbar_cb_destroy** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- int **ewl_statusbar_init** (Ewl_Statusbar *sb)
Initialize a statusbar to default values.
- void **ewl_statusbar_left_append** (Ewl_Statusbar *sb, Ewl_Widget *w)
pack the given widget w into the statusbar sb into right of the left box.
- void **ewl_statusbar_left_hide** (Ewl_Statusbar *sb)
hide the box on the left of the status bar
- void **ewl_statusbar_left_prepend** (Ewl_Statusbar *sb, Ewl_Widget *w)
pack the given widget w into the statusbar sb into the left of the left box
- void **ewl_statusbar_left_show** (Ewl_Statusbar *sb)
show the box on the left of the status bar
- Ewl_Widget * **ewl_statusbar_new** (void)
Allocate and initialize a new statusbar widget.
- void **ewl_statusbar_pop** (Ewl_Statusbar *sb)

Removes the most recent status message and displays the one before it.

- void **ewl_statusbar_push** (**Ewl_Statusbar** *sb, char *txt)

Displays the given status message on the statusbar.

- void **ewl_statusbar_right_append** (**Ewl_Statusbar** *sb, **Ewl_Widget** *w)

pack the given widget w into the statusbar sb into the right of the right box

- void **ewl_statusbar_right_hide** (**Ewl_Statusbar** *sb)

hide the box on the right of the status bar

- void **ewl_statusbar_right_prepend** (**Ewl_Statusbar** *sb, **Ewl_Widget** *w)

pack the given widget w into the statusbar sb into the left of the right box

- void **ewl_statusbar_right_show** (**Ewl_Statusbar** *sb)

show the box on the right of the status bar

8.191.1 Function Documentation

- 8.191.1.1 void **ewl_statusbar_cb_destroy** (**Ewl_Widget** * w, void *ev
__ *UNUSED* __, void *data __ *UNUSED* __)

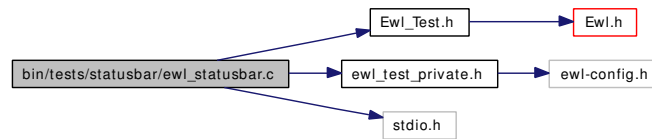
8.192 bin/tests/statusbar/ewl_statusbar.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_statusbar.c:



Functions

- `void test_info (Ewl_Test *test)`

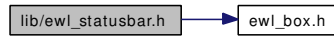
8.192.1 Function Documentation

8.192.1.1 `void test_info (Ewl_Test * test)`

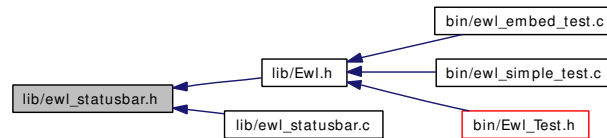
8.193 lib/ewl_statusbar.h File Reference

```
#include "ewl_box.h"
```

Include dependency graph for ewl_statusbar.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Statusbar**

*Inherits from the **Ewl_Box** (p. 586) class and extends it to provide for a simple status bar.*

Defines

- `#define EWL_STATUSBAR(statusbar) ((Ewl_Statusbar *) statusbar)`
- `#define EWL_STATUSBAR_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_STATUSBAR_TYPE))`
- `#define EWL_STATUSBAR_TYPE "statusbar"`

Typedefs

- `typedef Ewl_Statusbar Ewl_Statusbar`

Functions

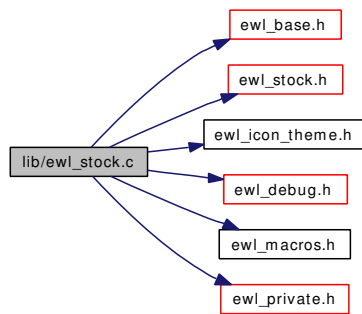
- `void ewl_statusbar_active_set (Ewl_Statusbar *sb, Ewl_Position pos)`
Sets the active statusbar container.
- `void ewl_statusbar_cb_destroy (Ewl_Widget *w, void *ev, void *data)`
- `int ewl_statusbar_init (Ewl_Statusbar *sb)`
Initialize a statusbar to default values.
- `void ewl_statusbar_left_append (Ewl_Statusbar *sb, Ewl_Widget *w)`
pack the given widget w into the statusbar sb into right of the left box.
- `void ewl_statusbar_left_hide (Ewl_Statusbar *sb)`
hide the box on the left of the status bar

- `void ewl_statusbar_left_prepend (Ewl_Statusbar *sb, Ewl_Widget *w)`
pack the given widget w into the statusbar sb into the left of the left box
- `void ewl_statusbar_left_show (Ewl_Statusbar *sb)`
show the box on the left of the status bar
- `Ewl_Widget * ewl_statusbar_new (void)`
Allocate and initialize a new statusbar widget.
- `void ewl_statusbar_pop (Ewl_Statusbar *sb)`
Removes the most recient status message and displays the one before it.
- `void ewl_statusbar_push (Ewl_Statusbar *sb, char *txt)`
Displays the given status message on the statusbar.
- `void ewl_statusbar_right_append (Ewl_Statusbar *sb, Ewl_Widget *w)`
pack the given widget w into the statusbar sb into the right of the right box
- `void ewl_statusbar_right_hide (Ewl_Statusbar *sb)`
hide the box on the right of the status bar
- `void ewl_statusbar_right_prepend (Ewl_Statusbar *sb, Ewl_Widget *w)`
pack the given widget w into the statusbar sb into the left of the right box
- `void ewl_statusbar_right_show (Ewl_Statusbar *sb)`
show the box on the right of the status bar

8.194 lib/ewl_stock.c File Reference

```
#include "ewl_base.h"
#include "ewl_stock.h"
#include "ewl_icon_theme.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_stock.c:



Functions

- void **ewl_stock_functions_set** (Ewl_Stock *s, Ewl_Stock_Funcs *funcs)
- int **ewl_stock_init** (Ewl_Stock *s)
Initialize the stock widget and inherited fields.
- Ewl_Stock_Type **ewl_stock_type_get** (Ewl_Stock *s)
Get the stock type in use.
- void **ewl_stock_type_set** (Ewl_Stock *s, Ewl_Stock_Type stock)
Set the type of the stock to use.

Variables

- struct {
 char * **image_key**
 char * **label**
 char * **tooltip**
} **ewl_stock_items** []

8.194.1 Variable Documentation

8.194.1.1 `struct { ... } ewl_stock_items[]`

8.194.1.2 `char* image_key`

8.194.1.3 `char* label`

8.194.1.4 `char* tooltip`

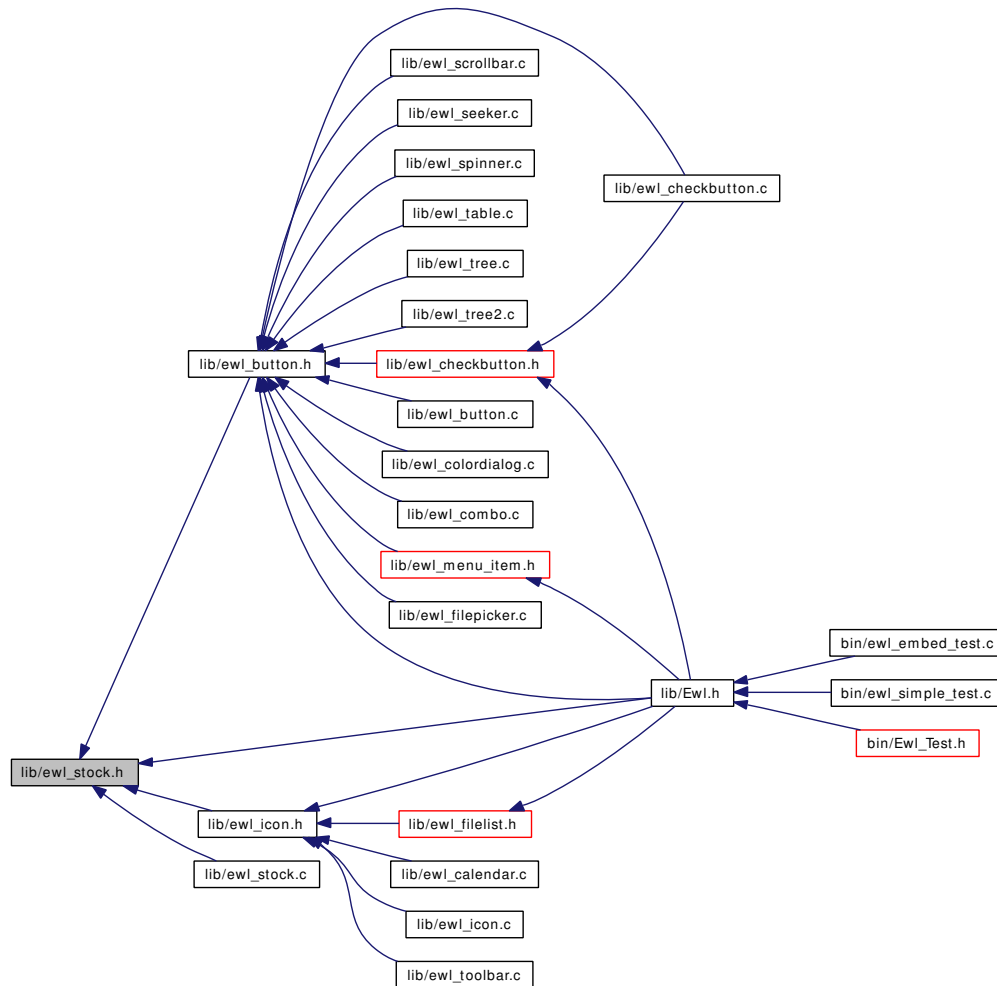
8.195 lib/ewl_stock.h File Reference

```
#include "ewl_box.h"
```

Include dependency graph for ewl_stock.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Stock**

*Inherits from **Ewl_Box** (p. 586) and provides the base for the other stock icons.*

- struct **Ewl_Stock_Funcs**

The function callbacks that an inheriting widget needs to implement.

Defines

- `#define EWL_STOCK(stock) ((Ewl_Stock *) stock)`
- `#define EWL_STOCK_IMAGE_SET(f) ((Ewl_Stock_Image_Set)f)`
- `#define EWL_STOCK_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_STOCK_TYPE))`
- `#define EWL_STOCK_LABEL_SET(f) ((Ewl_Stock_Label_Set)f)`
- `#define EWL_STOCK_TOOLTIP_SET(f) ((Ewl_Stock_Tooltip_Set)f)`
- `#define EWL_STOCK_TYPE "stock"`

Typedefs

- `typedef Ewl_Stock Ewl_Stock`
- `typedef Ewl_Stock_Funcs Ewl_Stock_Funcs`
- `typedef void(*) Ewl_Stock_Image_Set (Ewl_Stock *s, const char *file, const char *key)`
- `typedef void(*) Ewl_Stock_Label_Set (Ewl_Stock *s, const char *txt)`
- `typedef void(*) Ewl_Stock_Tooltip_Set (Ewl_Stock *s, const char *tip)`

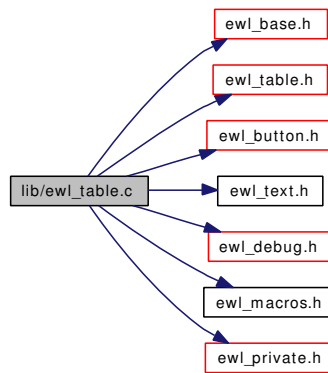
Functions

- `void ewl_stock_functions_set (Ewl_Stock *s, Ewl_Stock_Funcs *funcs)`
- `int ewl_stock_init (Ewl_Stock *s)`
Initialize the stock widget and inherited fields.
- `Ewl_Stock_Type ewl_stock_type_get (Ewl_Stock *s)`
Get the stock type in use.
- `void ewl_stock_type_set (Ewl_Stock *s, Ewl_Stock_Type stock)`
Set the type of the stock to use.

8.196 lib/ewl_table.c File Reference

```
#include "ewl_base.h"
#include "ewl_table.h"
#include "ewl_button.h"
#include "ewl_text.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_table.c:



Functions

- void **ewl_table_add** (**Ewl_Table** *table, **Ewl_Widget** *w, int start_col, int end_col, int start_row, int end_row)
Add a child widget to the table.
- void **ewl_table_cb_child_select** (**Ewl_Widget** *w, void *ev_data __UNUSED__, void *user_data)
- void **ewl_table_cb_child_show** (**Ewl_Container** *p, **Ewl_Widget** *c __UNUSED__)
- void **ewl_table_cb_configure** (**Ewl_Widget** *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_table_col_row_get** (**Ewl_Table** *t, **Ewl_Cell** *cell, int *start_col, int *end_col, int *start_row, int *end_row)
Get the column and row of a widget.
- void **ewl_table_column_w_get** (**Ewl_Table** *table, int col, int *width)
Get the width of a table column.
- void **ewl_table_column_w_set** (**Ewl_Table** *table, int col, int width)
Set the width of a table column.
- **Ecore_List** * **ewl_table_find** (**Ewl_Table** *t, int start_col, int end_col, int start_row, int end_row)

Get a list of the widgets in the specified col/row.

- unsigned int **ewl_table_hhomogeneous_get** (**Ewl_Table** *table)
Retrieves the horizontal homogeneous flag.
- void **ewl_table_hhomogeneous_set** (**Ewl_Table** *table, unsigned int h)
Change the horizontal homogeneous layout of the box.
- void **ewl_table_homogeneous_set** (**Ewl_Table** *table, unsigned int h)
Change the homogeneous layout of the box.
- int **ewl_table_init** (**Ewl_Table** *t, int cols, int rows, char **col_headers)
Initialize table to starting values.
- **Ewl_Widget** * **ewl_table_new** (int cols, int rows, char **col_headers)
Create a new table.
- void **ewl_table_reset** (**Ewl_Table** *t, int cols, int rows, char **col_headers)
Clear the table and set new geometry.
- void **ewl_table_row_h_get** (**Ewl_Table** *table, int row, int *height)
Get the height of a table row.
- void **ewl_table_row_h_set** (**Ewl_Table** *table, int row, int height)
Set the height of a table row.
- char * **ewl_table_selected_get** (**Ewl_Table** *t)
Get the text in the current selected box.
- unsigned int **ewl_table_vhomogeneous_get** (**Ewl_Table** *table)
Retrieves the vertical homogeneous flag.
- void **ewl_table_vhomogeneous_set** (**Ewl_Table** *table, unsigned int h)
Change the vertical homogeneous layout of the box.

8.196.1 Function Documentation

- 8.196.1.1 void **ewl_table_cb_child_select** (**Ewl_Widget** * *w*, void **ev_data* __ *UNUSED* __, void * *user_data*)
- 8.196.1.2 void **ewl_table_cb_child_show** (**Ewl_Container** * *p*, **Ewl_Widget** **c* __ *UNUSED* __)
- 8.196.1.3 void **ewl_table_cb_configure** (**Ewl_Widget** * *w*, void **ev_data* __ *UNUSED* __, void **user_data* __ *UNUSED* __)

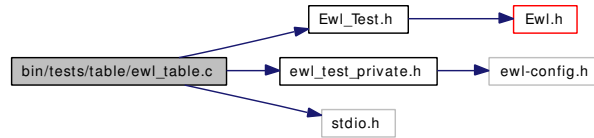
8.197 bin/tests/table/ewl_table.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_table.c:



Functions

- `void test_info (Ewl_Test *test)`

8.197.1 Function Documentation

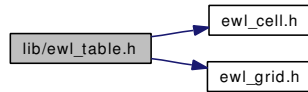
8.197.1.1 `void test_info (Ewl_Test * test)`

8.198 lib/ewl_table.h File Reference

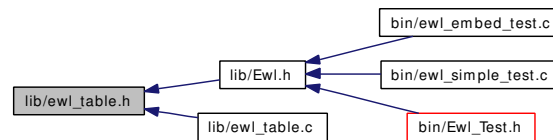
```
#include "ewl_cell.h"
```

```
#include "ewl_grid.h"
```

Include dependency graph for ewl_table.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Table**

*Inherits from **Ewl_Container** (p. 618) and extends to provide a table layout.*

Defines

- `#define EWL_TABLE(table) ((Ewl_Table *)table)`
- `#define EWL_TABLE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TABLE_TYPE))`
- `#define EWL_TABLE_TYPE "table"`

Typedefs

- `typedef Ewl_Table Ewl_Table`

Functions

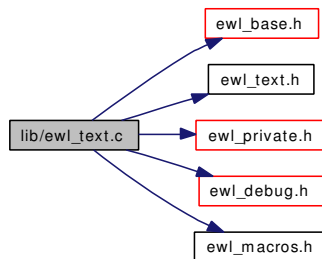
- `void ewl_table_add (Ewl_Table *table, Ewl_Widget *w, int start_col, int end_col, int start_row, int end_row)`
Add a child widget to the table.
- `void ewl_table_cb_child_select (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_table_cb_child_show (Ewl_Container *p, Ewl_Widget *c)`
- `void ewl_table_cb_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_table_col_row_get (Ewl_Table *table, Ewl_Cell *cell, int *start_col, int *end_col, int *start_row, int *end_row)`
Get the column and row of a widget.

- void **ewl_table_column_w_get** (**Ewl_Table** *table, int col, int *width)
Get the width of a table column.
- void **ewl_table_column_w_set** (**Ewl_Table** *table, int col, int width)
Set the width of a table column.
- **Ecore_List** * **ewl_table_find** (**Ewl_Table** *table, int start_col, int end_col, int start_row, int end_row)
Get a list of the widgets in the specified col/row.
- unsigned int **ewl_table_hhomogeneous_get** (**Ewl_Table** *table)
Retrieves the horizontal homogeneous flag.
- void **ewl_table_hhomogeneous_set** (**Ewl_Table** *table, unsigned int h)
Change the horizontal homogeneous layout of the box.
- void **ewl_table_homogeneous_set** (**Ewl_Table** *table, unsigned int h)
Change the homogeneous layout of the box.
- int **ewl_table_init** (**Ewl_Table** *t, int cols, int rows, char **col_headers)
Initialize table to starting values.
- **Ewl_Widget** * **ewl_table_new** (int cols, int rows, char **col_headers)
Create a new table.
- void **ewl_table_reset** (**Ewl_Table** *t, int cols, int rows, char **c_headers)
Clear the table and set new geometry.
- void **ewl_table_row_h_get** (**Ewl_Table** *table, int row, int *height)
Get the height of a table row.
- void **ewl_table_row_h_set** (**Ewl_Table** *table, int row, int height)
Set the height of a table row.
- char * **ewl_table_selected_get** (**Ewl_Table** *table)
Get the text in the current selected box.
- unsigned int **ewl_table_vhomogeneous_get** (**Ewl_Table** *table)
Retrieves the vertical homogeneous flag.
- void **ewl_table_vhomogeneous_set** (**Ewl_Table** *table, unsigned int h)
Change the vertical homogeneous layout of the box.

8.199 lib/ewl_text.c File Reference

```
#include "ewl_base.h"
#include "ewl_text.h"
#include "ewl_private.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
```

Include dependency graph for ewl_text.c:



Data Structures

- struct **Ewl_Text_Fmt**
Text formatting information.

Defines

- #define **COLOR_HASH**(r, g, b, a) (r << 24 | g << 16 | b << 8 | a)
- #define **EWL_TEXT_CHAR_BYTE_LEN**(s) (ewl_text_trailing_bytes[(unsigned int)(unsigned char)((s)[0])])
- #define **EWL_TEXT_EXTEND_VAL** 4096
- #define **KEY_COMPARE**(k1, k2) if (k1 > k2) goto CTX1_LARGER; else if (k2 > k1) goto CTX2_LARGER;

Typedefs

- typedef **Ewl_Text_Fmt** Ewl_Text_Fmt

Functions

- void **ewl_text_align_apply** (Ewl_Text *t, unsigned int align, unsigned int char_len)
This will set the given alignment from the current cursor position for the given length of text.
- unsigned int **ewl_text_align_get** (Ewl_Text *t, unsigned int char_idx)
Retrieves the alignment value from the given index.
- void **ewl_text_align_set** (Ewl_Text *t, unsigned int align)

Set the current alignment value of the text.

- void **ewl_text_bg_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)

This will set the bg colour of the text from the current cursor position to the given length.

- void **ewl_text_bg_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)

Get the text background colour at the given index.

- void **ewl_text_bg_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)

Set the background colour at the cursor.

- void **ewl_text_cb_child_add** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_text_cb_child_del** (Ewl_Container *c, Ewl_Widget *w, int idx __UNUSED__)
- void **ewl_text_cb_configure** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_text_cb_destroy** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_text_cb_hide** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_text_cb_mouse_down** (Ewl_Widget *w, void *ev, void *data __UNUSED__)
- void **ewl_text_cb_mouse_move** (Ewl_Widget *w, void *ev, void *data __UNUSED__)
- void **ewl_text_cb_mouse_up** (Ewl_Widget *w, void *ev, void *data __UNUSED__)
- void **ewl_text_cb_obscure** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_text_cb_reveal** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_text_cb_show** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_text_clear** (Ewl_Text *t)

Clear the text widget.

- void **ewl_text_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)

This will set the given colour from the current cursor position for the specified length.

- void **ewl_text_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)

Retrives the text colour at the given index.

- void **ewl_text_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)

Set the text colour at the cursor.

- void **ewl_text_context_acquire** (Ewl_Text_Context *tx)

- `int ewl_text_context_compare (Ewl_Text_Context *a, Ewl_Text_Context *b)`
- `Ewl_Text_Context * ewl_text_context_default_create (Ewl_Text *t)`
*Retrieves the default **Ewl_Text_Context** (p. 761).*
- `Ewl_Text_Context * ewl_text_context_dup (Ewl_Text_Context *old)`
- `int ewl_text_context_init (void)`
- `Ewl_Text_Context * ewl_text_context_new (void)`
- `void ewl_text_context_release (Ewl_Text_Context *tx)`
- `void ewl_text_context_shutdown (void)`
- `unsigned int ewl_text_coord_index_map (Ewl_Text *t, int x, int y)`
Map the given coordinate into an index into the text widget.
- `unsigned int ewl_text_cursor_position_get (Ewl_Text *t)`
Retrieve the cursor position from the text widget.
- `unsigned int ewl_text_cursor_position_line_down_get (Ewl_Text *t)`
Get the index if we were to move the cursor down one line.
- `unsigned int ewl_text_cursor_position_line_up_get (Ewl_Text *t)`
Get the index if we were to move the cursor up one line.
- `void ewl_text_cursor_position_set (Ewl_Text *t, unsigned int char_pos)`
Set the cursor position in the text widget.
- `void ewl_text_double_underline_color_apply (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)`
This will set the double_underline colour of the text from the current cursor position to the given length.
- `void ewl_text_double_underline_color_get (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)`
Retrieve the double underline colour at the given index.
- `void ewl_text_double_underline_color_set (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)`
Set the double underline colour at the cursor.
- `void ewl_text_fmt_dump (Ewl_Text *t)`
Prints out the formatting information for the given text object.
- `void ewl_text_font_apply (Ewl_Text *t, const char *font, unsigned int char_len)`
This will apply the specified font from the current cursor position to the length specified.
- `char * ewl_text_font_get (Ewl_Text *t, unsigned int char_idx)`
This will retrieve the font used at the specified index in the text.
- `void ewl_text_font_set (Ewl_Text *t, const char *font)`
This will set the current font to be used when we insert more text.
- `void ewl_text_font_size_apply (Ewl_Text *t, unsigned int size, unsigned int char_len)`

This will apply the font size to the text from the current cursor position for the given length.

- unsigned int **ewl_text_font_size_get** (Ewl_Text *t, unsigned int char_idx)
Retrieve the font size at the given index.
- void **ewl_text_font_size_set** (Ewl_Text *t, unsigned int size)
Set the font size to use when inserting new text.
- void **ewl_text_font_source_apply** (Ewl_Text *t, const char *source, const char *font, unsigned int char_len)
This will apply the specified font from the current cursor position to the length specified.
- char * **ewl_text_font_source_get** (Ewl_Text *t, unsigned int char_idx)
This will retrieve the font source used at the specified index in the text.
- void **ewl_text_font_source_set** (Ewl_Text *t, const char *source, const char *font)
This will set the current font to be used when we insert more text.
- void **ewl_text_glow_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)
This will set the glow colour of the text from the current cursor position to the given length.
- void **ewl_text_glow_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)
Get the glow colour at the given index.
- void **ewl_text_glow_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)
Set the glow colour at the cursor.
- unsigned int **ewl_text_has_selection** (Ewl_Text *t)
Check if anything is selected in the text widget.
- void **ewl_text_index_geometry_map** (Ewl_Text *t, unsigned int char_idx, int *x, int *y, int *w, int *h)
Map the given character index into a position in the text widget.
- int **ewl_text_init** (Ewl_Text *t)
*Initializes an **Ewl_Text** (p. 757) widget to default values.*
- unsigned int **ewl_text_length_get** (Ewl_Text *t)
Retrieve the character length of the text.
- Ewl_Widget * **ewl_text_new** (void)
*Creates a new **Ewl_Text** (p. 757) widget.*
- void **ewl_text_offsets_get** (Ewl_Text *t, int *x, int *y)
Retrieve the current layout offsets of the text.
- void **ewl_text_offsets_set** (Ewl_Text *t, int x, int y)

Set the current layout offsets of the text.

- void **ewl_text_outline_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)

This will set the outline colour of the text from the current cursor position to the given length.

- void **ewl_text_outline_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)

Get the outline colour at the given index.

- void **ewl_text_outline_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)

Set the outline colour at the cursor.

- unsigned int **ewl_text_selectable_get** (Ewl_Text *t)

Get the selectable state of the text.

- void **ewl_text_selectable_set** (Ewl_Text *t, unsigned int selectable)

Set if the text is selectable.

- Ewl_Text_Trigger * **ewl_text_selection_get** (Ewl_Text *t)

Get the current text selection.

- char * **ewl_text_selection_text_get** (Ewl_Text *t)

Gets the current text of the selection.

- void **ewl_text_shadow_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)

This will set the shadow colour of the text from the current cursor position to the given length.

- void **ewl_text_shadow_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)

Retrieve the shadow colour at the given index.

- void **ewl_text_shadow_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)

Set the shadow colour at the cursor.

- void **ewl_text_strikethrough_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)

This will set the strikethrough colour of the text from the current cursor position to the given length.

- void **ewl_text_strikethrough_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)

Retrieve the strikethrough colour at the given index.

- void **ewl_text_strikethrough_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)

Set the strikethrough colour at the cursor.

- void **ewl_text_style_add** (Ewl_Text *t, Ewl_Text_Style style, unsigned int char_len)
This will add the given style to the text from the cursor up to length characters.
- void **ewl_text_style_del** (Ewl_Text *t, Ewl_Text_Style style, unsigned int char_len)
This will delete the given style from the text starting at the cursor up to length characters.
- unsigned int **ewl_text_style_has** (Ewl_Text *t, Ewl_Text_Style style, unsigned int char_idx)
Check if the given style is set at the given index in the text.
- void **ewl_text_style_invert** (Ewl_Text *t, Ewl_Text_Style style, unsigned int char_len)
This will invert the given style in the text starting at the cursor up to length characters.
- void **ewl_text_styles_apply** (Ewl_Text *t, unsigned int styles, unsigned int char_len)
This will set the given style from the current cursor position for the given length of text.
- unsigned int **ewl_text_styles_get** (Ewl_Text *t, unsigned int char_idx)
Retrives the styles in use at the given index.
- void **ewl_text_styles_set** (Ewl_Text *t, unsigned int styles)
Sets the given styles into the text at the cursor.
- void **ewl_text_text_append** (Ewl_Text *t, const char *text)
Append the text into the text widget.
- void **ewl_text_text_delete** (Ewl_Text *t, unsigned int char_len)
This will delete the specified length of text from the current cursor position.
- char * **ewl_text_text_get** (Ewl_Text *t)
Retrieve the text from the text widget.
- void **ewl_text_text_insert** (Ewl_Text *t, const char *text, unsigned int char_idx)
Insert the given text into the text widget.
- void **ewl_text_text_prepend** (Ewl_Text *t, const char *text)
Prepend the given text into the text widget.
- void **ewl_text_text_set** (Ewl_Text *t, const char *text)
Set the text in the text widget.
- int **ewl_text_trigger_area_init** (Ewl_Text_Trigger_Area *area, Ewl_Text_Trigger_Type type)
- Ewl_Widget * **ewl_text_trigger_area_new** (Ewl_Text_Trigger_Type type)
- unsigned int **ewl_text_trigger_base_get** (Ewl_Text_Trigger *t)
Retrieves the current base position of the cursor.

- void **ewl_text_trigger_base_set** (Ewl_Text_Trigger *t, unsigned int char_pos)

Sets the given position pos as the base for the trigger t.
- void **ewl_text_trigger_cb_destroy** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_text_trigger_cb_mouse_down** (Ewl_Widget *w __UNUSED__, void *ev, void *data)
- void **ewl_text_trigger_cb_mouse_in** (Ewl_Widget *w __UNUSED__, void *ev, void *data)
- void **ewl_text_trigger_cb_mouse_out** (Ewl_Widget *w __UNUSED__, void *ev, void *data)
- void **ewl_text_trigger_cb_mouse_up** (Ewl_Widget *w __UNUSED__, void *ev, void *data)
- int **ewl_text_trigger_init** (Ewl_Text_Trigger *trigger, Ewl_Text_Trigger_Type type)

Initializes a trigger to default values.
- unsigned int **ewl_text_trigger_length_get** (Ewl_Text_Trigger *t)

Retrieves the length from the cursor t.
- void **ewl_text_trigger_length_set** (Ewl_Text_Trigger *t, unsigned int char_len)

Sets the length len on the trigger t.
- Ewl_Text_Trigger * **ewl_text_trigger_new** (Ewl_Text_Trigger_Type type)

Creates a new trigger for the text object.
- unsigned int **ewl_text_trigger_start_pos_get** (Ewl_Text_Trigger *t)

Retrieves the start position of the trigger.
- void **ewl_text_trigger_start_pos_set** (Ewl_Text_Trigger *t, unsigned int char_pos)

Sets the start position of the trigger t to position pos.
- Ewl_Text_Trigger_Type **ewl_text_trigger_type_get** (Ewl_Text_Trigger *t)

Retrieves the type of the trigger.
- void **ewl_text_triggers_configure** (Ewl_Text *t)
- void **ewl_text_triggers_hide** (Ewl_Text *t)
- void **ewl_text_triggers_realize** (Ewl_Text *t)
- void **ewl_text_triggers_show** (Ewl_Text *t)
- void **ewl_text_triggers_unrealize** (Ewl_Text *t)
- void **ewl_text_underline_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)

This will set the underline colour of the text from the current cursor position to the given length.
- void **ewl_text_underline_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)

Retrieve the underline colour at the given index.

- void **ewl_text_underline_color_set** (**Ewl_Text** *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)

Set the underline colour at the cursor.

- void **ewl_text_wrap_apply** (**Ewl_Text** *t, **Ewl_Text_Wrap** wrap, unsigned int char_len)

This will apply the given wrap value from the current cursor position for the given length of text.

- **Ewl_Text_Wrap** **ewl_text_wrap_get** (**Ewl_Text** *t, unsigned int char_idx)

Retrives the text wrap value at the given index.

- void **ewl_text_wrap_set** (**Ewl_Text** *t, **Ewl_Text_Wrap** wrap)

Sets the wrap value of the text at the given index.

8.199.1 Define Documentation

8.199.1.1 **#define** **COLOR_HASH**(r, g, b, a) (r << 24 | g << 16 | b << 8 | a)

8.199.1.2 **#define** **EWL_TEXT_CHAR_BYTE_LEN**(s) (ewl_text_trailing_bytes[(unsigned int)(unsigned char)((s)[0])])

8.199.1.3 **#define** **EWL_TEXT_EXTEND_VAL** 4096

8.199.1.4 **#define** **KEY_COMPARE**(k1, k2) if (k1 > k2) goto CTX1_LARGER;
else if (k2 > k1) goto CTX2_LARGER;

8.199.2 Typedef Documentation

8.199.2.1 **typedef struct** **Ewl_Text_Fmt** **Ewl_Text_Fmt**

Ewl_Text_Fmt (p. 764)

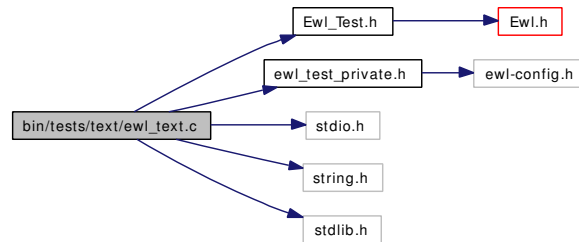
8.199.3 Function Documentation

- 8.199.3.1 void ewl_text_cb_configure (Ewl_Widget * *w*, void *ev __ *UNUSED* __, void *data __ *UNUSED* __)
- 8.199.3.2 void ewl_text_cb_destroy (Ewl_Widget * *w*, void *ev __ *UNUSED* __, void *data __ *UNUSED* __)
- 8.199.3.3 void ewl_text_cb_hide (Ewl_Widget * *w*, void *ev __ *UNUSED* __, void *data __ *UNUSED* __)
- 8.199.3.4 void ewl_text_cb_mouse_down (Ewl_Widget * *w*, void * *ev*, void *data __ *UNUSED* __)
- 8.199.3.5 void ewl_text_cb_mouse_move (Ewl_Widget * *w*, void * *ev*, void *data __ *UNUSED* __)
- 8.199.3.6 void ewl_text_cb_mouse_up (Ewl_Widget * *w*, void * *ev*, void *data __ *UNUSED* __)
- 8.199.3.7 void ewl_text_cb_obscure (Ewl_Widget * *w*, void *ev __ *UNUSED* __, void *data __ *UNUSED* __)
- 8.199.3.8 void ewl_text_cb_reveal (Ewl_Widget * *w*, void *ev __ *UNUSED* __, void *data __ *UNUSED* __)
- 8.199.3.9 void ewl_text_cb_show (Ewl_Widget * *w*, void *ev __ *UNUSED* __, void *data __ *UNUSED* __)
- 8.199.3.10 void ewl_text_trigger_cb_destroy (Ewl_Widget * *w*, void *ev_data __ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.199.3.11 void ewl_text_trigger_cb_mouse_down (Ewl_Widget **w* __ *UNUSED* __, void * *ev*, void * *data*)
- 8.199.3.12 void ewl_text_trigger_cb_mouse_in (Ewl_Widget **w* __ *UNUSED* __, void * *ev*, void * *data*)
- 8.199.3.13 void ewl_text_trigger_cb_mouse_out (Ewl_Widget **w* __ *UNUSED* __, void * *ev*, void * *data*)
- 8.199.3.14 void ewl_text_trigger_cb_mouse_up (Ewl_Widget **w* __ *UNUSED* __, void * *ev*, void * *data*)

8.200 bin/tests/text/ewl_text.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

Include dependency graph for ewl_text.c:



Functions

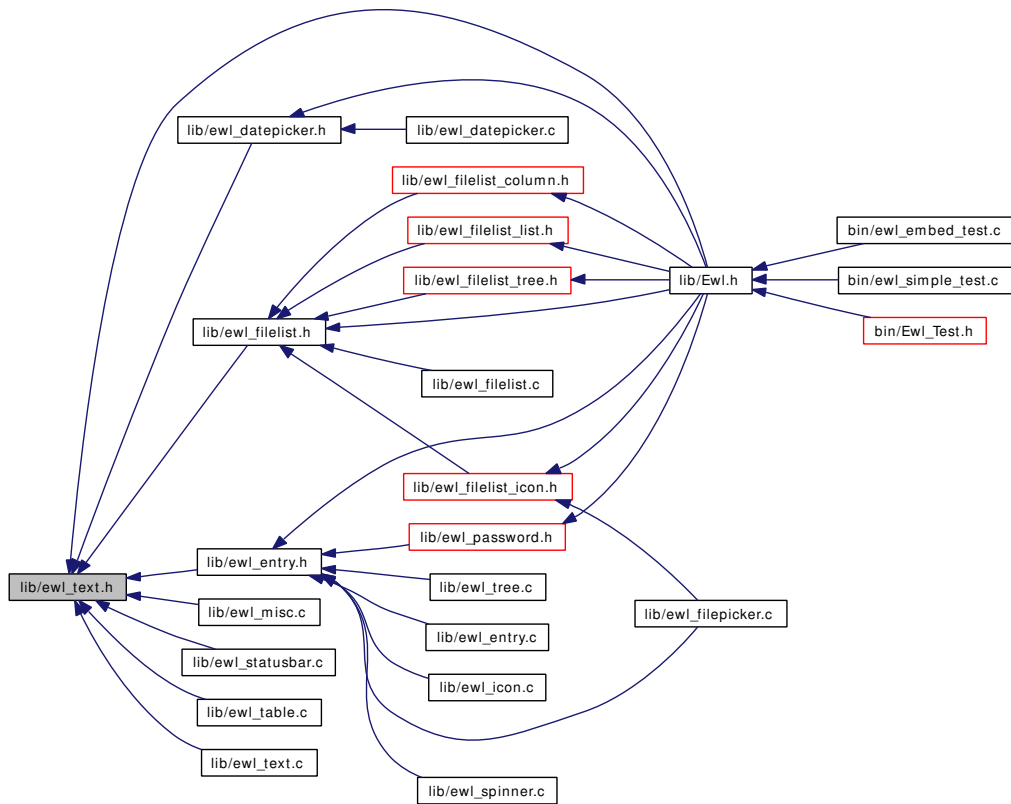
- `void test_info (Ewl_Test *test)`

8.200.1 Function Documentation

8.200.1.1 `void test_info (Ewl_Test * test)`

8.201 lib/ewl_text.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Text**
*Inherits from the **Ewl_Container** (p. 618) class and extends it to provide text layout and formatting.*
- struct **Ewl_Text_Context**
Stores context information for the different formatting nodes.
- struct **Ewl_Text_Trigger**
*Inherits from **Ewl_Widget** (p. 785) and extends to provide a trigger for the text widget.*
- struct **Ewl_Text_Trigger_Area**
*Inherits from **Ewl_Widget** (p. 785) and extends to provide a trigger area.*

Defines

- `#define EWL_TEXT(x) ((Ewl_Text *)x)`

- `#define EWL_TEXT_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TEXT_TYPE))`
- `#define EWL_TEXT_SELECTION_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TEXT_SELECTION_TYPE))`
- `#define EWL_TEXT_SELECTION_TYPE "selection"`
- `#define EWL_TEXT_TRIGGER(trigger) ((Ewl_Text_Trigger *) trigger)`
- `#define EWL_TEXT_TRIGGER_AREA(area) ((Ewl_Text_Trigger_Area *) area)`
- `#define EWL_TEXT_TRIGGER_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TEXT_TRIGGER_TYPE))`
- `#define EWL_TEXT_TRIGGER_TYPE "trigger"`
- `#define EWL_TEXT_TYPE "text"`

Typedefs

- `typedef Ewl_Text Ewl_Text`
- `typedef Ewl_Text_Context Ewl_Text_Context`
- `typedef Ewl_Text_Trigger Ewl_Text_Trigger`
- `typedef Ewl_Text_Trigger_Area Ewl_Text_Trigger_Area`

Functions

- `void ewl_text_align_apply (Ewl_Text *t, unsigned int align, unsigned int char_len)`
This will set the given alignment from the current cursor position for the given length of text.
- `unsigned int ewl_text_align_get (Ewl_Text *t, unsigned int char_idx)`
Retrieves the alignment value from the given index.
- `void ewl_text_align_set (Ewl_Text *t, unsigned int align)`
Set the current alignment value of the text.
- `void ewl_text_bg_color_apply (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)`
This will set the bg colour of the text from the current cursor position to the given length.
- `void ewl_text_bg_color_get (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)`
Get the text background colour at the given index.
- `void ewl_text_bg_color_set (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)`
Set the background colour at the cursor.
- `void ewl_text_cb_child_add (Ewl_Container *c, Ewl_Widget *w)`
- `void ewl_text_cb_child_del (Ewl_Container *c, Ewl_Widget *w, int idx)`
- `void ewl_text_cb_configure (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_text_cb_destroy (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_text_cb_hide (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_text_cb_mouse_down (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_text_cb_mouse_move (Ewl_Widget *w, void *ev, void *data)`

- void **ewl_text_cb_mouse_up** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_cb_obscure** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_cb_reveal** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_cb_show** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_text_clear** (Ewl_Text *t)

Clear the text widget.
- void **ewl_text_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)

This will set the given colour from the current cursor position for the specified length.
- void **ewl_text_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)

Retrives the text colour at the given index.
- void **ewl_text_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)

Set the text colour at the cursor.
- void **ewl_text_context_acquire** (Ewl_Text_Context *tx)
- int **ewl_text_context_compare** (Ewl_Text_Context *a, Ewl_Text_Context *b)
- Ewl_Text_Context * **ewl_text_context_default_create** (Ewl_Text *t)

*Retrieves the default **Ewl_Text_Context** (p. 761).*
- Ewl_Text_Context * **ewl_text_context_dup** (Ewl_Text_Context *old)
- int **ewl_text_context_init** (void)
- Ewl_Text_Context * **ewl_text_context_new** (void)
- void **ewl_text_context_release** (Ewl_Text_Context *tx)
- void **ewl_text_context_shutdown** (void)
- unsigned int **ewl_text_coord_index_map** (Ewl_Text *t, int x, int y)

Map the given coordinate into an index into the text widget.
- unsigned int **ewl_text_cursor_position_get** (Ewl_Text *t)

Retrieve the cursor position from the text widget.
- unsigned int **ewl_text_cursor_position_line_down_get** (Ewl_Text *t)

Get the index if we were to move the cursor down one line.
- unsigned int **ewl_text_cursor_position_line_up_get** (Ewl_Text *t)

Get the index if we were to move the cursor up one line.
- void **ewl_text_cursor_position_set** (Ewl_Text *t, unsigned int char_pos)

Set the cursor position in the text widget.
- void **ewl_text_double_underline_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)

This will set the double_underline colour of the text from the current cursor position to the given length.
- void **ewl_text_double_underline_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)

Retrieve the double underline colour at the given index.

- void **ewl_text_double_underline_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)

Set the double underline colour at the cursor.

- void **ewl_text_fmt_dump** (Ewl_Text *t)

Prints out the formatting information for the given text object.

- void **ewl_text_font_apply** (Ewl_Text *t, const char *font, unsigned int char_len)

This will apply the specified font from the current cursor position to the length specified.

- char * **ewl_text_font_get** (Ewl_Text *t, unsigned int char_idx)

This will retrieve the font used at the specified index in the text.

- void **ewl_text_font_set** (Ewl_Text *t, const char *font)

This will set the current font to be used when we insert more text.

- void **ewl_text_font_size_apply** (Ewl_Text *t, unsigned int size, unsigned int char_len)

This will apply the font size to the text from the current cursor position for the given length.

- unsigned int **ewl_text_font_size_get** (Ewl_Text *t, unsigned int char_idx)

Retrieve the font size at the given index.

- void **ewl_text_font_size_set** (Ewl_Text *t, unsigned int size)

Set the font size to use when inserting new text.

- void **ewl_text_font_source_apply** (Ewl_Text *t, const char *source, const char *font, unsigned int char_len)

This will apply the specified font from the current cursor position to the length specified.

- char * **ewl_text_font_source_get** (Ewl_Text *t, unsigned int char_idx)

This will retrieve the font source used at the specified index in the text.

- void **ewl_text_font_source_set** (Ewl_Text *t, const char *source, const char *font)

This will set the current font to be used when we insert more text.

- void **ewl_text_glow_color_apply** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)

This will set the glow colour of the text from the current cursor position to the given length.

- void **ewl_text_glow_color_get** (Ewl_Text *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)

Get the glow colour at the given index.

- void **ewl_text_glow_color_set** (Ewl_Text *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)

Set the glow colour at the cursor.

- unsigned int **ewl_text_has_selection** (**Ewl_Text** *t)
Check if anything is selected in the text widget.
- void **ewl_text_index_geometry_map** (**Ewl_Text** *t, unsigned int char_idx, int *x, int *y, int *w, int *h)
Map the given character index into a position in the text widget.
- int **ewl_text_init** (**Ewl_Text** *t)
*Initializes an **Ewl_Text** (p. 757) widget to default values.*
- unsigned int **ewl_text_length_get** (**Ewl_Text** *t)
Retrieve the character length of the text.
- **Ewl_Widget** * **ewl_text_new** (void)
*Creates a new **Ewl_Text** (p. 757) widget.*
- void **ewl_text_offsets_get** (**Ewl_Text** *t, int *x, int *y)
Retrieve the current layout offsets of the text.
- void **ewl_text_offsets_set** (**Ewl_Text** *t, int x, int y)
Set the current layout offsets of the text.
- void **ewl_text_outline_color_apply** (**Ewl_Text** *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)
This will set the outline colour of the text from the current cursor position to the given length.
- void **ewl_text_outline_color_get** (**Ewl_Text** *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)
Get the outline colour at the given index.
- void **ewl_text_outline_color_set** (**Ewl_Text** *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)
Set the outline colour at the cursor.
- unsigned int **ewl_text_selectable_get** (**Ewl_Text** *t)
Get the selectable state of the text.
- void **ewl_text_selectable_set** (**Ewl_Text** *t, unsigned int selectable)
Set if the text is selectable.
- **Ewl_Text_Trigger** * **ewl_text_selection_get** (**Ewl_Text** *t)
Get the current text selection.
- char * **ewl_text_selection_text_get** (**Ewl_Text** *t)
Gets the current text of the selection.
- void **ewl_text_shadow_color_apply** (**Ewl_Text** *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)
This will set the shadow colour of the text from the current cursor position to the given length.

- void **ewl_text_shadow_color_get** (**Ewl_Text** *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)
Retrieve the shadow colour at the given index.
- void **ewl_text_shadow_color_set** (**Ewl_Text** *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)
Set the shadow colour at the cursor.
- void **ewl_text_strikethrough_color_apply** (**Ewl_Text** *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)
This will set the strikethrough colour of the text from the current cursor position to the given length.
- void **ewl_text_strikethrough_color_get** (**Ewl_Text** *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)
Retrieve the strikethrough colour at the given index.
- void **ewl_text_strikethrough_color_set** (**Ewl_Text** *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)
Set the strikethrough colour at the cursor.
- void **ewl_text_style_add** (**Ewl_Text** *t, **Ewl_Text_Style** style, unsigned int char_len)
This will add the given style to the text from the cursor up to length characters.
- void **ewl_text_style_del** (**Ewl_Text** *t, **Ewl_Text_Style** style, unsigned int char_len)
This will delete the given style from the text starting at the cursor up to length characters.
- unsigned int **ewl_text_style_has** (**Ewl_Text** *t, **Ewl_Text_Style** style, unsigned int char_idx)
Check if the given style is set at the given index in the text.
- void **ewl_text_style_invert** (**Ewl_Text** *t, **Ewl_Text_Style** style, unsigned int char_len)
This will invert the given style in the text starting at the cursor up to length characters.
- void **ewl_text_styles_apply** (**Ewl_Text** *t, unsigned int styles, unsigned int char_len)
This will set the given style from the current cursor position for the given length of text.
- unsigned int **ewl_text_styles_get** (**Ewl_Text** *t, unsigned int char_idx)
Retrives the styles in use at the given index.
- void **ewl_text_styles_set** (**Ewl_Text** *t, unsigned int styles)
Sets the given styles into the text at the cursor.
- void **ewl_text_text_append** (**Ewl_Text** *t, const char *text)
Append the text into the text widget.
- void **ewl_text_text_delete** (**Ewl_Text** *t, unsigned int length)

This will delete the specified length of text from the current cursor position.

- `char * ewl_text_text_get (Ewl_Text *t)`
Retrieve the text from the text widget.
- `void ewl_text_text_insert (Ewl_Text *t, const char *text, unsigned int char_idx)`
Insert the given text into the text widget.
- `void ewl_text_text_prepend (Ewl_Text *t, const char *text)`
Prepend the given text into the text widget.
- `void ewl_text_text_set (Ewl_Text *t, const char *text)`
Set the text in the text widget.
- `int ewl_text_trigger_area_init (Ewl_Text_Trigger_Area *area, Ewl_Text_Trigger_Type type)`
- `Ewl_Widget * ewl_text_trigger_area_new (Ewl_Text_Trigger_Type type)`
- `unsigned int ewl_text_trigger_base_get (Ewl_Text_Trigger *t)`
Retrieves the current base position of the cursor.
- `void ewl_text_trigger_base_set (Ewl_Text_Trigger *t, unsigned int char_pos)`
Sets the given position pos as the base for the trigger t.
- `void ewl_text_trigger_cb_destroy (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_text_trigger_cb_mouse_down (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_text_trigger_cb_mouse_in (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_text_trigger_cb_mouse_out (Ewl_Widget *w, void *ev, void *data)`
- `void ewl_text_trigger_cb_mouse_up (Ewl_Widget *w, void *ev, void *data)`
- `int ewl_text_trigger_init (Ewl_Text_Trigger *trigger, Ewl_Text_Trigger_Type type)`
Initializes a trigger to default values.
- `unsigned int ewl_text_trigger_length_get (Ewl_Text_Trigger *t)`
Retrieves the length from the cursor t.
- `void ewl_text_trigger_length_set (Ewl_Text_Trigger *t, unsigned int char_len)`
Sets the length len on the trigger t.
- `Ewl_Text_Trigger * ewl_text_trigger_new (Ewl_Text_Trigger_Type type)`
Creates a new trigger for the text object.
- `unsigned int ewl_text_trigger_start_pos_get (Ewl_Text_Trigger *t)`
Retrieves the start position of the trigger.
- `void ewl_text_trigger_start_pos_set (Ewl_Text_Trigger *t, unsigned int char_pos)`
Sets the start position of the trigger t to position pos.
- `Ewl_Text_Trigger_Type ewl_text_trigger_type_get (Ewl_Text_Trigger *t)`

Retrieves the type of the trigger.

- void **ewl_text_triggers_configure** (**Ewl_Text** *t)
- void **ewl_text_triggers_hide** (**Ewl_Text** *t)
- void **ewl_text_triggers_realize** (**Ewl_Text** *t)
- void **ewl_text_triggers_show** (**Ewl_Text** *t)
- void **ewl_text_triggers_unrealize** (**Ewl_Text** *t)
- void **ewl_text_underline_color_apply** (**Ewl_Text** *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a, unsigned int char_len)

This will set the underline colour of the text from the current cursor position to the given length.

- void **ewl_text_underline_color_get** (**Ewl_Text** *t, unsigned int *r, unsigned int *g, unsigned int *b, unsigned int *a, unsigned int char_idx)

Retrieve the underline colour at the given index.

- void **ewl_text_underline_color_set** (**Ewl_Text** *t, unsigned int r, unsigned int g, unsigned int b, unsigned int a)

Set the underline colour at the cursor.

- void **ewl_text_wrap_apply** (**Ewl_Text** *t, **Ewl_Text_Wrap** wrap, unsigned int char_len)

This will apply the given wrap value from the current cursor position for the given length of text.

- **Ewl_Text_Wrap** **ewl_text_wrap_get** (**Ewl_Text** *t, unsigned int char_idx)

Retrives the text wrap value at the given index.

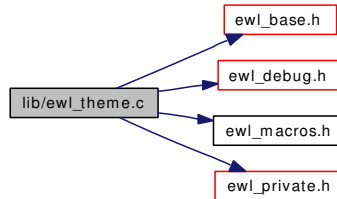
- void **ewl_text_wrap_set** (**Ewl_Text** *t, **Ewl_Text_Wrap** wrap)

Sets the wrap value of the text at the given index.

8.202 lib/ewl_theme.c File Reference

```
#include "ewl_base.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_theme.c:



Functions

- **int ewl_theme_data_int_get** (Ewl_Widget *w, char *k)
Retrieve an integer value from a widgets theme.
- **void ewl_theme_data_int_set** (Ewl_Widget *w, char *k, int v)
Store data into a widgets theme.
- **void ewl_theme_data_reset** (Ewl_Widget *w)
Reset the theme settings for the widget w.
- **const char * ewl_theme_data_str_get** (Ewl_Widget *w, char *k)
Retrieve an string value from a widgets theme.
- **void ewl_theme_data_str_set** (Ewl_Widget *w, char *k, char *v)
Store data into a widgets theme.
- **void ewl_theme_font_path_add** (char *path)
Add a specified path to the font search path.
- **Ecore_List * ewl_theme_font_path_get** (void)
retrieve the path of a widgets theme's fonts
- **char * ewl_theme_image_get** (Ewl_Widget *w, char *k)
retrieve the path to an image from a widgets theme
- **int ewl_theme_init** (void)
- **const char * ewl_theme_path_get** (void)
Return the path of the current theme.
- **void ewl_theme_shutdown** (void)
- **int ewl_theme_theme_set** (const char *theme_name)

- `int ewl_theme_widget_init (Ewl_Widget *w)`
- `void ewl_theme_widget_shutdown (Ewl_Widget *w)`

Variables

- `Ecore_List * ewl_embed_list`

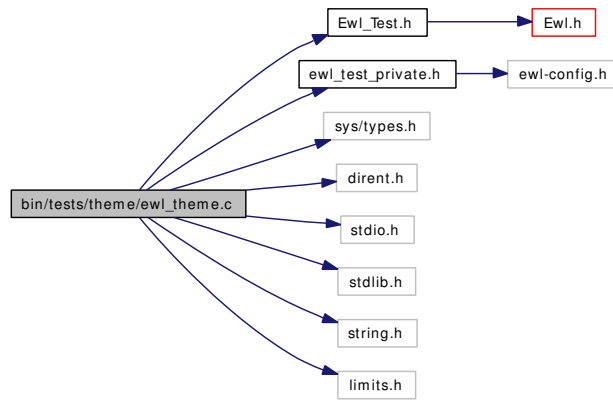
8.202.1 Variable Documentation

8.202.1.1 Ecore_List* ewl_embed_list

8.203 bin/tests/theme/ewl_theme.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <sys/types.h>
#include <dirent.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <limits.h>
```

Include dependency graph for ewl_theme.c:



Functions

- `void test_info (Ewl_Test *test)`

8.203.1 Function Documentation

8.203.1.1 `void test_info (Ewl_Test * test)`

8.204 lib/ewl_theme.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- `#define EWL_THEME_KEY_NOMATCH ((char *)0xdeadbeef)`

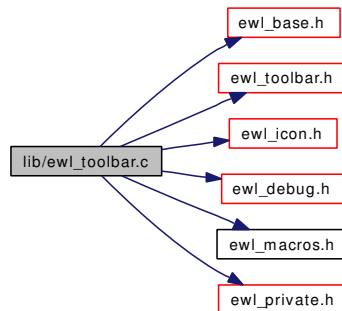
Functions

- `int ewl_theme_data_int_get (Ewl_Widget *w, char *k)`
Retrieve an integer value from a widgets theme.
- `void ewl_theme_data_int_set (Ewl_Widget *w, char *k, int v)`
Store data into a widgets theme.
- `void ewl_theme_data_reset (Ewl_Widget *w)`
Reset the theme settings for the widget w.
- `const char * ewl_theme_data_str_get (Ewl_Widget *w, char *k)`
Retrieve an string value from a widgets theme.
- `void ewl_theme_data_str_set (Ewl_Widget *w, char *k, char *v)`
Store data into a widgets theme.
- `void ewl_theme_font_path_add (char *path)`
Add a specified path to the font search path.
- `Ecore_List * ewl_theme_font_path_get (void)`
retrieve the path of a widgets theme's fonts
- `char * ewl_theme_image_get (Ewl_Widget *w, char *k)`
retrieve the path to an image from a widgets theme
- `int ewl_theme_init (void)`
- `const char * ewl_theme_path_get (void)`
Return the path of the current theme.
- `void ewl_theme_shutdown (void)`
- `int ewl_theme_theme_set (const char *theme)`
- `int ewl_theme_widget_init (Ewl_Widget *w)`
- `void ewl_theme_widget_shutdown (Ewl_Widget *w)`

8.205 lib/ewl_toolbar.c File Reference

```
#include "ewl_base.h"
#include "ewl_toolbar.h"
#include "ewl_icon.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_toolbar.c:



Functions

- **Ewl_Widget * ewl_htoolbar_new** (void)
Allocate a new toolbar widget with horizontal orientation.
- void **ewl_toolbar_cb_child_add** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_toolbar_icon_part_hide** (Ewl_Toolbar *t, Ewl_Icon_Part part)
Set the orientation of the toolbar.
- int **ewl_toolbar_init** (Ewl_Toolbar *t)
Initialize the toolbar and inherited fields.
- **Ewl_Widget * ewl_toolbar_new** (void)
Allocate a new toolbar widget with default (horizontal) orientation.
- **Ewl_Orientation ewl_toolbar_orientation_get** (Ewl_Toolbar *t)
This retrieves the current orientation of the toolbar.
- void **ewl_toolbar_orientation_set** (Ewl_Toolbar *t, Ewl_Orientation o)
Set the orientation of the toolbar.
- **Ewl_Widget * ewl_vtoolbar_new** (void)
Allocate a new toolbar widget with vertical orientation.

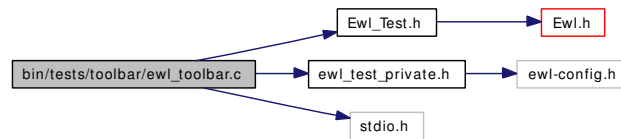
8.206 bin/tests/toolbar/ewl_toolbar.c File Reference

```
#include "Ewl_Test.h"
```

```
#include "ewl_test_private.h"
```

```
#include <stdio.h>
```

Include dependency graph for ewl_toolbar.c:



Functions

- void `cb_click` (`Ewl_Widget *w`, void *e `__UNUSED__`, void *data)
- void `test_info` (`Ewl_Test *test`)

8.206.1 Function Documentation

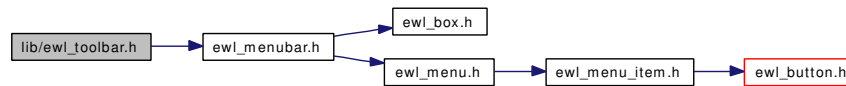
8.206.1.1 void `cb_click` (`Ewl_Widget * w`, void *e `__UNUSED__`, void * *data*)

8.206.1.2 void `test_info` (`Ewl_Test * test`)

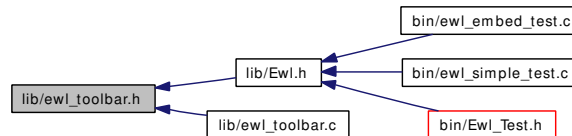
8.207 lib/ewl_toolbar.h File Reference

```
#include "ewl_menubar.h"
```

Include dependency graph for ewl_toolbar.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Toolbar**

*Inherits from **Ewl_Menubar** (p. 701) and extends to provide a toolbar layout.*

Defines

- `#define EWL_TOOLBAR(toolbar) ((Ewl_Toolbar *) toolbar)`
- `#define EWL_TOOLBAR_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TOOLBAR_TYPE))`
- `#define EWL_TOOLBAR_TYPE "toolbar"`

Typedefs

- `typedef Ewl_Toolbar Ewl_Toolbar`

Functions

- **Ewl_Widget *** `ewl_htoolbar_new` (void)
Allocate a new toolbar widget with horizontal orientation.
- void `ewl_toolbar_cb_child_add` (Ewl_Container *c, Ewl_Widget *w)
- void `ewl_toolbar_icon_part_hide` (Ewl_Toolbar *t, Ewl_Icon_Part part)
Set the orientation of the toolbar.
- int `ewl_toolbar_init` (Ewl_Toolbar *t)
Initialize the toolbar and inherited fields.
- **Ewl_Widget *** `ewl_toolbar_new` (void)

Allocate a new toolbar widget with default (horizontal) orientation.

- **Ewl_Orientation ewl_toolbar_orientation_get (Ewl_Toolbar *t)**

This retrieves the current orientation of the toolbar.

- **void ewl_toolbar_orientation_set (Ewl_Toolbar *t, Ewl_Orientation o)**

Set the orientation of the toolbar.

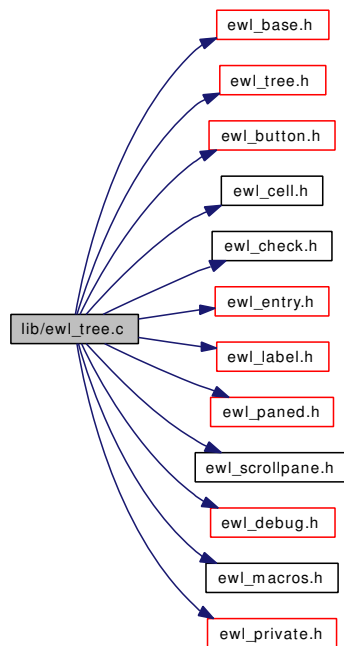
- **Ewl_Widget * ewl_vtoolbar_new (void)**

Allocate a new toolbar widget with vertical orientation.

8.208 lib/ewl_tree.c File Reference

```
#include "ewl_base.h"
#include "ewl_tree.h"
#include "ewl_button.h"
#include "ewl_cell.h"
#include "ewl_check.h"
#include "ewl_entry.h"
#include "ewl_label.h"
#include "ewl_paned.h"
#include "ewl_scrollpane.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_tree.c:



Functions

- void **ewl_tree_cb_child_resize** (Ewl_Container *c)
- void **ewl_tree_cb_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_tree_cb_destroy** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_tree_cb_header_change** (Ewl_Widget *w __UNUSED__, void *ev __UNUSED__, void *data)

- void **ewl_tree_cb_header_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_tree_cb_hscroll** (Ewl_Widget *w __UNUSED__, void *ev_data __UNUSED__, void *user_data)
- void **ewl_tree_cb_node_child_add** (Ewl_Container *c, Ewl_Widget *w __UNUSED__)
- void **ewl_tree_cb_node_child_del** (Ewl_Container *c, Ewl_Widget *w, int idx __UNUSED__)
- void **ewl_tree_cb_node_child_hide** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_tree_cb_node_child_show** (Ewl_Container *c, Ewl_Widget *w __UNUSED__)
- void **ewl_tree_cb_node_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_tree_cb_node_destroy** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_tree_cb_node_resize** (Ewl_Container *c, Ewl_Widget *w, int size __UNUSED__, Ewl_Orientation o __UNUSED__)
- void **ewl_tree_cb_node_toggle** (Ewl_Widget *w __UNUSED__, void *ev_data __UNUSED__, void *user_data)
- void **ewl_tree_cb_row_hide** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_tree_cb_row_select** (Ewl_Widget *w, void *ev_data, void *user_data __UNUSED__)
- void **ewl_tree_columns_set** (Ewl_Tree *tree, unsigned short columns)

Change the number of columns displayed in a tree.

- **Ewl_Widget* ewl_tree_entry_row_add** (Ewl_Tree *tree, Ewl_Row *prow, char **text)

Add a row of text entries to a tree.

- unsigned int **ewl_tree_expandable_rows_get** (Ewl_Tree *tree)

Gets the header visibility of the tree.

- void **ewl_tree_expandable_rows_set** (Ewl_Tree *tree, unsigned int expand)

Sets the expandability of the rows in the tree.

- void **ewl_tree_headers_set** (Ewl_Tree *tree, char **headers)

Change the widgets in a trees column headers.

- unsigned int **ewl_tree_headers_visible_get** (Ewl_Tree *tree)

Gets the header visibility of the tree.

- void **ewl_tree_headers_visible_set** (Ewl_Tree *tree, unsigned int visible)

Sets the visibility of the headers in the tree.

- int **ewl_tree_init** (Ewl_Tree *tree, unsigned short columns)

Initialize the contents of a tree widget.

- **Ewl_Selection_Mode ewl_tree_mode_get** (Ewl_Tree *tree)

Retrieve the current selection mode of a tree.

- void **ewl_tree_mode_set** (**Ewl_Tree** *tree, **Ewl_Selection_Mode** mode)
Change the selection mode for a specified tree.
- **Ewl_Widget** * **ewl_tree_new** (unsigned short columns)
Allocate and initialize a new tree widget.
- void **ewl_tree_node_collapse** (**Ewl_Tree_Node** *node)
Collapse a node in the tree.
- void **ewl_tree_node_expand** (**Ewl_Tree_Node** *node)
Expand a node in the tree.
- int **ewl_tree_node_expandable_get** (**Ewl_Tree_Node** *node)
Retrieves the expand setting from the node.
- void **ewl_tree_node_expandable_set** (**Ewl_Tree_Node** *node, int expand)
Sets the expand value into the node.
- int **ewl_tree_node_init** (**Ewl_Tree_Node** *node)
Initialize the node fields of an inheriting object.
- **Ewl_Widget** * **ewl_tree_node_new** ()
Allocate and initialize a new node.
- **Ewl_Widget** * **ewl_tree_row_add** (**Ewl_Tree** *tree, **Ewl_Row** *prow, **Ewl_Widget** **children)
Add a group of widgets to a row in the tree.
- **Ewl_Widget** * **ewl_tree_row_column_get** (**Ewl_Row** *row, int i)
Retrieves the actual widget added via row_add instead of the cell.
- void **ewl_tree_row_destroy** (**Ewl_Tree** *tree, **Ewl_Row** *row)
Destroy a specified row from the tree.
- void **ewl_tree_row_expand_set** (**Ewl_Row** *row, **Ewl_Tree_Node_Flags** expanded)
Set the expand state of a specific row.
- **Ewl_Widget** * **ewl_tree_row_find** (**Ewl_Tree** *tree, int row)
- void **ewl_tree_row_remove** (**Ewl_Tree** *tree, **Ewl_Row** *row)
Remove a specified row from the tree.
- void **ewl_tree_selected_clear** (**Ewl_Tree** *tree)
Clear the current selection from a tree.
- **Ecore_List** * **ewl_tree_selected_get** (**Ewl_Tree** *tree)
Retrieves a list of selected rows from a tree.
- **Ewl_Widget** * **ewl_tree_text_row_add** (**Ewl_Tree** *tree, **Ewl_Row** *prow, char **text)
Add a row of text to a tree.

8.208.1 Function Documentation

- 8.208.1.1 `void ewl_tree_cb_configure (Ewl_Widget * w, void *ev_data
__ UNUSED __, void *user_data __ UNUSED __)`
- 8.208.1.2 `void ewl_tree_cb_destroy (Ewl_Widget * w, void *ev_data
__ UNUSED __, void *user_data __ UNUSED __)`
- 8.208.1.3 `void ewl_tree_cb_header_change (Ewl_Widget *w __ UNUSED __,
void *ev __ UNUSED __, void * data)`
- 8.208.1.4 `void ewl_tree_cb_header_configure (Ewl_Widget * w, void *ev_data
__ UNUSED __, void *user_data __ UNUSED __)`
- 8.208.1.5 `void ewl_tree_cb_hscroll (Ewl_Widget *w __ UNUSED __, void
*ev_data __ UNUSED __, void * user_data)`
- 8.208.1.6 `void ewl_tree_cb_node_child_add (Ewl_Container * c, Ewl_Widget
*w __ UNUSED __)`
- 8.208.1.7 `void ewl_tree_cb_node_child_show (Ewl_Container * c, Ewl_Widget
*w __ UNUSED __)`
- 8.208.1.8 `void ewl_tree_cb_node_configure (Ewl_Widget * w, void *ev_data
__ UNUSED __, void *user_data __ UNUSED __)`
- 8.208.1.9 `void ewl_tree_cb_node_destroy (Ewl_Widget * w, void *ev_data
__ UNUSED __, void *user_data __ UNUSED __)`
- 8.208.1.10 `void ewl_tree_cb_node_toggle (Ewl_Widget *w __ UNUSED __,
void *ev_data __ UNUSED __, void * user_data)`
- 8.208.1.11 `void ewl_tree_cb_row_hide (Ewl_Widget * w, void *ev_data
__ UNUSED __, void *user_data __ UNUSED __)`
- 8.208.1.12 `void ewl_tree_cb_row_select (Ewl_Widget * w, void * ev_data, void
*user_data __ UNUSED __)`
- 8.208.1.13 `void ewl_tree_row_remove (Ewl_Tree * tree, Ewl_Row * row)`

Remove a specified row from the tree.

Parameters:

tree,: the tree to remove a row from

row,: the row to be removed from the tree

Returns:

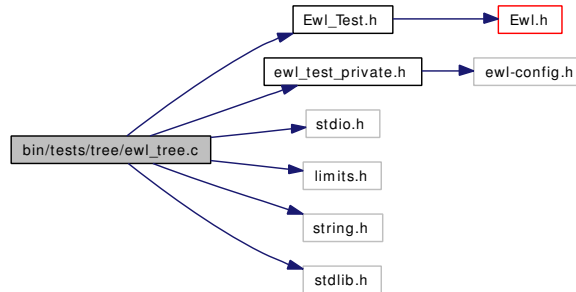
Returns no value.

Removes *row* from *tree* if it is present in *tree*. The widgets in the row will not be destroyed, so they can be accessed at a later time.

8.209 bin/tests/tree/ewl_tree.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <limits.h>
#include <string.h>
#include <stdlib.h>
```

Include dependency graph for ewl_tree.c:



Defines

- `#define COLS 4`
- `#define NEST 3`
- `#define ROWS 50`

Functions

- `void test_info (Ewl_Test *test)`

8.209.1 Define Documentation

8.209.1.1 `#define COLS 4`

8.209.1.2 `#define NEST 3`

8.209.1.3 `#define ROWS 50`

8.209.2 Function Documentation

8.209.2.1 `void test_info (Ewl_Test * test)`

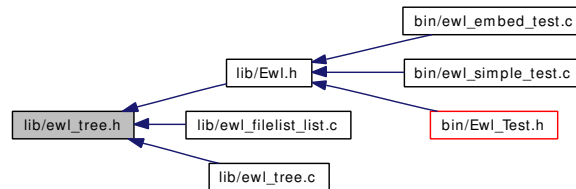
8.210 lib/ewl_tree.h File Reference

```
#include "ewl_row.h"
```

Include dependency graph for ewl_tree.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Tree**
*Inherits from **Ewl_Container** (p. 618) and extends to provide a tree layout.*
- struct **Ewl_Tree_Node**
*Inherits from **Ewl_Container** (p. 618) and extends to hold information on a row of the tree.*

Defines

- `#define EWL_TREE(t) ((Ewl_Tree *)t)`
- `#define EWL_TREE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE_TYPE))`
- `#define EWL_TREE_NODE(t) ((Ewl_Tree_Node *)t)`
- `#define EWL_TREE_NODE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE_NODE_TYPE))`
- `#define EWL_TREE_NODE_TYPE "node"`
- `#define EWL_TREE_TYPE "tree"`

Typedefs

- `typedef Ewl_Tree Ewl_Tree`
- `typedef Ewl_Tree_Node Ewl_Tree_Node`

Functions

- `void ewl_tree_cb_child_resize (Ewl_Container *c)`
- `void ewl_tree_cb_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_tree_cb_destroy (Ewl_Widget *w, void *ev_data, void *user_data)`

- void **ewl_tree_cb_header_change** (Ewl_Widget *w, void *ev, void *data)
- void **ewl_tree_cb_header_configure** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_tree_cb_hscroll** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_tree_cb_node_child_add** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_tree_cb_node_child_del** (Ewl_Container *c, Ewl_Widget *w, int idx)
- void **ewl_tree_cb_node_child_hide** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_tree_cb_node_child_show** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_tree_cb_node_configure** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_tree_cb_node_destroy** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_tree_cb_node_resize** (Ewl_Container *c, Ewl_Widget *w, int size, Ewl_Orientation o)
- void **ewl_tree_cb_node_toggle** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_tree_cb_row_hide** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_tree_cb_row_select** (Ewl_Widget *w, void *ev_data, void *user_data)
- void **ewl_tree_columns_set** (Ewl_Tree *tree, unsigned short columns)

Change the number of columns displayed in a tree.

- **Ewl_Widget *ewl_tree_entry_row_add** (Ewl_Tree *tree, Ewl_Row *prow, char **text)

Add a row of text entries to a tree.

- unsigned int **ewl_tree_expandable_rows_get** (Ewl_Tree *tree)

Gets the header visibility of the tree.

- void **ewl_tree_expandable_rows_set** (Ewl_Tree *tree, unsigned int visible)

Sets the expandability of the rows in the tree.

- void **ewl_tree_headers_set** (Ewl_Tree *tree, char **headers)

Change the widgets in a trees column headers.

- unsigned int **ewl_tree_headers_visible_get** (Ewl_Tree *tree)

Gets the header visibility of the tree.

- void **ewl_tree_headers_visible_set** (Ewl_Tree *tree, unsigned int visible)

Sets the visibility of the headers in the tree.

- int **ewl_tree_init** (Ewl_Tree *tree, unsigned short columns)

Initialize the contents of a tree widget.

- **Ewl_Selection_Mode ewl_tree_mode_get** (Ewl_Tree *tree)

Retrieve the current selection mode of a tree.

- void **ewl_tree_mode_set** (Ewl_Tree *tree, Ewl_Selection_Mode mode)

Change the selection mode for a specified tree.

- **Ewl_Widget *ewl_tree_new** (unsigned short columns)

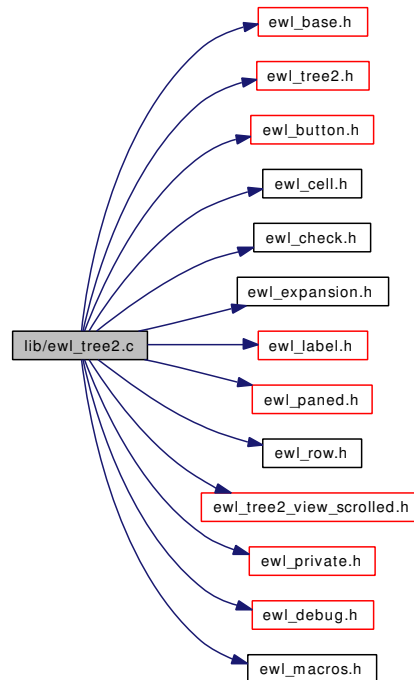
Allocate and initialize a new tree widget.

- void **ewl_tree_node_collapse** (**Ewl_Tree_Node** *tree)
Collapse a node in the tree.
- void **ewl_tree_node_expand** (**Ewl_Tree_Node** *tree)
Expand a node in the tree.
- int **ewl_tree_node_expandable_get** (**Ewl_Tree_Node** *node)
Retrieves the expand setting from the node.
- void **ewl_tree_node_expandable_set** (**Ewl_Tree_Node** *node, int expand)
Sets the expand value into the node.
- int **ewl_tree_node_init** (**Ewl_Tree_Node** *tree_node)
Initialize the node fields of an inheriting object.
- **Ewl_Widget** * **ewl_tree_node_new** (void)
Allocate and initialize a new node.
- **Ewl_Widget** * **ewl_tree_row_add** (**Ewl_Tree** *tree, **Ewl_Row** *prow, **Ewl_Widget** **children)
Add a group of widgets to a row in the tree.
- **Ewl_Widget** * **ewl_tree_row_column_get** (**Ewl_Row** *row, int i)
Retrieves the actual widget added via row_add instead of the cell.
- void **ewl_tree_row_destroy** (**Ewl_Tree** *tree, **Ewl_Row** *row)
Destroy a specified row from the tree.
- void **ewl_tree_row_expand_set** (**Ewl_Row** *row, **Ewl_Tree_Node_Flags** expanded)
Set the expand state of a specific row.
- **Ewl_Widget** * **ewl_tree_row_find** (**Ewl_Tree** *tree, int row)
- void **ewl_tree_selected_clear** (**Ewl_Tree** *tree)
Clear the current selection from a tree.
- **Ecore_List** * **ewl_tree_selected_get** (**Ewl_Tree** *tree)
Retrieves a list of selected rows from a tree.
- **Ewl_Widget** * **ewl_tree_text_row_add** (**Ewl_Tree** *tree, **Ewl_Row** *prow, char **text)
Add a row of text to a tree.

8.211 lib/ewl_tree2.c File Reference

```
#include "ewl_base.h"
#include "ewl_tree2.h"
#include "ewl_button.h"
#include "ewl_cell.h"
#include "ewl_check.h"
#include "ewl_expansion.h"
#include "ewl_label.h"
#include "ewl_paned.h"
#include "ewl_row.h"
#include "ewl_tree2_view_scrolled.h"
#include "ewl_private.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
```

Include dependency graph for ewl_tree2.c:



Functions

- void **ewl_tree2_cb_column_sort** (Ewl_Widget *w, void *ev __UNUSED__, void *data)
- void **ewl_tree2_cb_configure** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)

- void **ewl_tree2_cb_destroy** (Ewl_Widget *w, void *ev __UNUSED__, void *data __UNUSED__)
- void **ewl_tree2_cb_node_child_add** (Ewl_Container *c, Ewl_Widget *w __UNUSED__)
- void **ewl_tree2_cb_node_child_del** (Ewl_Container *c, Ewl_Widget *w, int idx __UNUSED__)
- void **ewl_tree2_cb_node_child_hide** (Ewl_Container *c, Ewl_Widget *w)
- void **ewl_tree2_cb_node_child_show** (Ewl_Container *c, Ewl_Widget *w __UNUSED__)
- void **ewl_tree2_cb_node_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_tree2_cb_node_resize** (Ewl_Container *c, Ewl_Widget *w, int size __UNUSED__, Ewl_Orientation o __UNUSED__)
- void **ewl_tree2_cb_node_toggle** (Ewl_Widget *w __UNUSED__, void *ev_data __UNUSED__, void *data)
- unsigned int **ewl_tree2_column_count_get** (Ewl_Tree2 *tree)
Retrives the number of columns in the tree.
- void **ewl_tree2_column_count_set** (Ewl_Tree2 *tree, unsigned int count)
Sets the number of columns in the tree.
- Ewl_View * **ewl_tree2_content_view_get** (Ewl_Tree2 *tree)
Retrives the view used to generate the tree content area.
- void **ewl_tree2_content_view_set** (Ewl_Tree2 *tree, Ewl_View *view)
Sets the view to use to generate the content area.
- Ewl_Widget * **ewl_tree2_content_widget_get** (Ewl_Tree2 *tree)
Retrieves the widget containing the tree rows.
- unsigned int **ewl_tree2_fixed_rows_get** (Ewl_Tree2 *tree)
Retrieve the fixed row size of the tree.
- void **ewl_tree2_fixed_rows_set** (Ewl_Tree2 *tree, unsigned int fixed)
Set the fixed row size of the tree.
- unsigned int **ewl_tree2_headers_visible_get** (Ewl_Tree2 *tree)
Retrieve if the header is visible in the tree.
- void **ewl_tree2_headers_visible_set** (Ewl_Tree2 *tree, unsigned char visible)
Toggle if the header is visible in the tree.
- int **ewl_tree2_init** (Ewl_Tree2 *tree)
Initialize the contents of a tree widget.
- Ewl_Widget * **ewl_tree2_new** (void)
Allocate and initialize a new tree widget.
- void **ewl_tree2_node_collapse** (Ewl_Tree2_Node *node)
- void **ewl_tree2_node_expand** (Ewl_Tree2_Node *node)

- unsigned int `ewl_tree2_node_expandable_get` (Ewl_Tree2_Node *node)
- void `ewl_tree2_node_expandable_set` (Ewl_Tree2_Node *node, void *data)
- unsigned int `ewl_tree2_node_expanded_is` (Ewl_Tree2_Node *node)
- int `ewl_tree2_node_init` (Ewl_Tree2_Node *node)
- Ewl_Widget * `ewl_tree2_node_new` (void)
- void `ewl_tree2_row_collapse` (Ewl_Tree2 *tree, void *data, unsigned int row)
Sets the given row to collapsed for the given data in tree.
- void `ewl_tree2_row_expand` (Ewl_Tree2 *tree, void *data, unsigned int row)
When the tree displays the data in data it will expand the give row. This data is the parent of the expansion row.
- unsigned int `ewl_tree2_row_expanded_is` (Ewl_Tree2 *tree, void *data, unsigned int row)
Checks if row is expanded in data of tree.
- Ewl_Tree_Selection_Type `ewl_tree2_selection_type_get` (Ewl_Tree2 *tree)
Get the selection type from the tree.
- void `ewl_tree2_selection_type_set` (Ewl_Tree2 *tree, Ewl_Tree_Selection_Type type)
Set the mode of the tree.

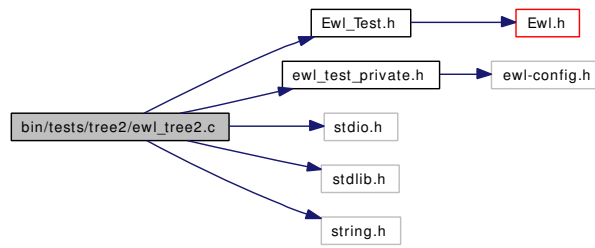
8.211.1 Function Documentation

- 8.211.1.1 void `ewl_tree2_cb_column_sort` (Ewl_Widget * *w*, void *ev __ *UNUSED* __, void * *data*)
- 8.211.1.2 void `ewl_tree2_cb_configure` (Ewl_Widget * *w*, void *ev __ *UNUSED* __, void *data __ *UNUSED* __)
- 8.211.1.3 void `ewl_tree2_cb_destroy` (Ewl_Widget * *w*, void *ev __ *UNUSED* __, void *data __ *UNUSED* __)
- 8.211.1.4 void `ewl_tree2_cb_node_child_add` (Ewl_Container * *c*, Ewl_Widget *w __ *UNUSED* __)
- 8.211.1.5 void `ewl_tree2_cb_node_child_show` (Ewl_Container * *c*, Ewl_Widget *w __ *UNUSED* __)
- 8.211.1.6 void `ewl_tree2_cb_node_configure` (Ewl_Widget * *w*, void *ev_data __ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.211.1.7 void `ewl_tree2_cb_node_toggle` (Ewl_Widget *w __ *UNUSED* __, void *ev_data __ *UNUSED* __, void * *data*)

8.212 bin/tests/tree2/ewl_tree2.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for ewl_tree2.c:



Data Structures

- struct `Tree2_Test_Data`
- struct `Tree2_Test_Row_Data`

Defines

- `#define TREE2_DATA_ELEMENTS 5`

Typedefs

- `typedef Tree2_Test_Data Tree2_Test_Data`
- `typedef Tree2_Test_Row_Data Tree2_Test_Row_Data`

Functions

- void `test_info (Ewl_Test *test)`

8.212.1 Define Documentation

8.212.1.1 `#define TREE2_DATA_ELEMENTS 5`

8.212.2 Typedef Documentation

8.212.2.1 `typedef struct Tree2_Test_Data Tree2_Test_Data`

8.212.2.2 `typedef struct Tree2_Test_Row_Data Tree2_Test_Row_Data`

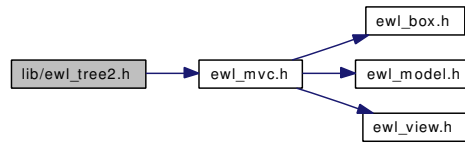
8.212.3 Function Documentation

8.212.3.1 `void test_info (Ewl_Test * test)`

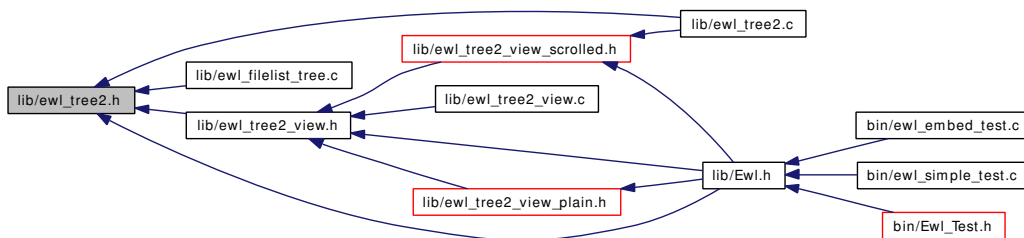
8.213 lib/ewl_tree2.h File Reference

```
#include "ewl_mvc.h"
```

Include dependency graph for ewl_tree2.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Tree2**
*Inherits from **Ewl_MVC** (p. 705) and extends to provide a tree widget.*
- struct **Ewl_Tree2_Node**
*Inherits from **Ewl_Container** (p. 618) and extends to hold information on a row of the tree.*

Defines

- `#define EWL_TREE2(t) ((Ewl_Tree2 *)t)`
- `#define EWL_TREE2_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE2_TYPE))`
- `#define EWL_TREE2_NODE(n) ((Ewl_Tree2_Node *)n)`
- `#define EWL_TREE2_NODE_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE2_NODE_TYPE))`
- `#define EWL_TREE2_NODE_TYPE "node"`
- `#define EWL_TREE2_TYPE "tree2"`

Typedefs

- `typedef Ewl_Tree2 Ewl_Tree2`
- `typedef Ewl_Tree2_Node Ewl_Tree2_Node`

Functions

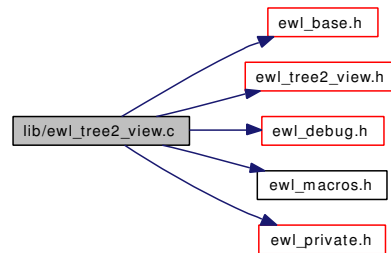
- void **ewl_tree2_cb_column_sort** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_tree2_cb_configure** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_tree2_cb_destroy** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_tree2_cb_node_child_add** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_tree2_cb_node_child_del** (**Ewl_Container** *c, **Ewl_Widget** *w, int idx)
- void **ewl_tree2_cb_node_child_hide** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_tree2_cb_node_child_show** (**Ewl_Container** *c, **Ewl_Widget** *w)
- void **ewl_tree2_cb_node_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_tree2_cb_node_realize** (**Ewl_Widget** *w, void *ev, void *data)
- void **ewl_tree2_cb_node_resize** (**Ewl_Container** *c, **Ewl_Widget** *w, int size, **Ewl_Orientation** o)
- void **ewl_tree2_cb_node_toggle** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- unsigned int **ewl_tree2_column_count_get** (**Ewl_Tree2** *tree)
Retrives the number of columns in the tree.
- void **ewl_tree2_column_count_set** (**Ewl_Tree2** *tree, unsigned int count)
Sets the number of columns in the tree.
- **Ewl_View** * **ewl_tree2_content_view_get** (**Ewl_Tree2** *tree)
Retrives the view used to generate the tree content area.
- void **ewl_tree2_content_view_set** (**Ewl_Tree2** *tree, **Ewl_View** *view)
Sets the view to use to generate the content area.
- **Ewl_Widget** * **ewl_tree2_content_widget_get** (**Ewl_Tree2** *tree)
Retrieves the widget containing the tree rows.
- unsigned int **ewl_tree2_fixed_rows_get** (**Ewl_Tree2** *tree)
Retrieve the fixed row size of the tree.
- void **ewl_tree2_fixed_rows_set** (**Ewl_Tree2** *tree, unsigned int fixed)
Set the fixed row size of the tree.
- unsigned int **ewl_tree2_headers_visible_get** (**Ewl_Tree2** *tree)
Retrieve if the header is visible in the tree.
- void **ewl_tree2_headers_visible_set** (**Ewl_Tree2** *tree, unsigned char visible)
Toggle if the header is visible in the tree.
- int **ewl_tree2_init** (**Ewl_Tree2** *tree)
Initialize the contents of a tree widget.
- **Ewl_Widget** * **ewl_tree2_new** (void)
Allocate and initialize a new tree widget.
- void **ewl_tree2_node_collapse** (**Ewl_Tree2_Node** *node)
- void **ewl_tree2_node_expand** (**Ewl_Tree2_Node** *node)

- unsigned int **ewl_tree2_node_expandable_get** (**Ewl_Tree2_Node** *node)
- void **ewl_tree2_node_expandable_set** (**Ewl_Tree2_Node** *node, void *data)
- unsigned int **ewl_tree2_node_expanded_is** (**Ewl_Tree2_Node** *node)
- int **ewl_tree2_node_init** (**Ewl_Tree2_Node** *node)
- **Ewl_Widget** * **ewl_tree2_node_new** (void)
- void **ewl_tree2_row_collapse** (**Ewl_Tree2** *tree, void *data, unsigned int row)
Sets the given row to collapsed for the given data in tree.
- void **ewl_tree2_row_expand** (**Ewl_Tree2** *tree, void *data, unsigned int row)
When the tree displays the data in data it will expand the give row. This data is the parent of the expansion row.
- unsigned int **ewl_tree2_row_expanded_is** (**Ewl_Tree2** *tree, void *data, unsigned int row)
Checks if row is expanded in data of tree.
- **Ewl_Tree_Selection_Type** **ewl_tree2_selection_type_get** (**Ewl_Tree2** *tree)
Get the selection type from the tree.
- void **ewl_tree2_selection_type_set** (**Ewl_Tree2** *tree, **Ewl_Tree_Selection_Type** type)
Set the mode of the tree.

8.214 lib/ewl_tree2_view.c File Reference

```
#include "ewl_base.h"
#include "ewl_tree2_view.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_tree2_view.c:



Functions

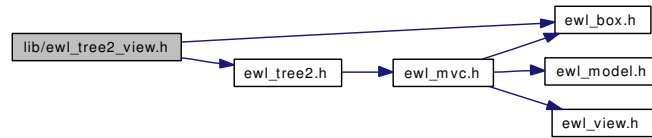
- `int ewl_tree2_view_init (Ewl_Tree2_View *v)`
*Initializes an **Ewl_Tree2_View** (p. 776) to default values.*
- `Ewl_Tree2 * ewl_tree2_view_tree2_get (Ewl_Tree2_View *v)`
Retrieves the tree parent set into the view.
- `void ewl_tree2_view_tree2_set (Ewl_Tree2_View *v, Ewl_Tree2 *t)`
*Sets the **t** tree into the **v** view.*

8.215 lib/ewl_tree2_view.h File Reference

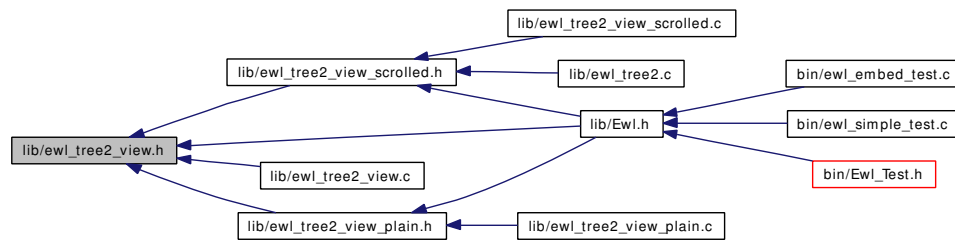
```
#include "ewl_box.h"
```

```
#include "ewl_tree2.h"
```

Include dependency graph for ewl_tree2_view.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Tree2_View**

*Inherits from **Ewl_Box** (p. 586) and provides the base information for a tree view.*

Defines

- #define **EWL_TREE2_VIEW(v)** ((**Ewl_Tree2_View** *)v)
- #define **EWL_TREE2_VIEW_IS(w)** (ewl_widget_type_is(**EWL_WIDGET(w)**, **EWL_TREE2_VIEW_TYPE**))
- #define **EWL_TREE2_VIEW_TYPE** "tree2_view"

Typedefs

- typedef **Ewl_Tree2_View** **Ewl_Tree2_View**

Functions

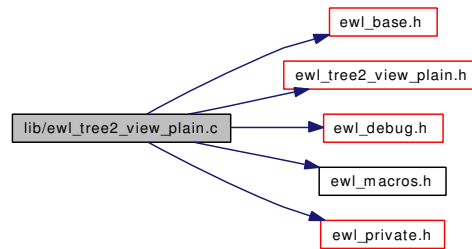
- int **ewl_tree2_view_init** (**Ewl_Tree2_View** *v)
*Initializes an **Ewl_Tree2_View** (p. 776) to default values.*
- **Ewl_Tree2 *** **ewl_tree2_view_tree2_get** (**Ewl_Tree2_View** *v)
Retrieves the tree parent set into the view.

- `void ewl_tree2_view_tree2_set (Ewl_Tree2_View *v, Ewl_Tree2 *t)`
Sets the t tree into the v view.

8.216 lib/ewl_tree2_view_plain.c File Reference

```
#include "ewl_base.h"
#include "ewl_tree2_view_plain.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_tree2_view_plain.c:



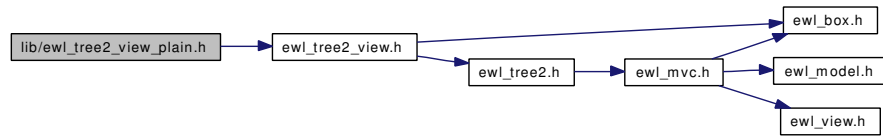
Functions

- **Ewl_View * ewl_tree2_view_plain_get (void)**
*Retrieves a shared **Ewl_Tree2_View_Plain** (p. 778) widget.*
- **int ewl_tree2_view_plain_init (Ewl_Tree2_View_Plain *tv)**
*Initializes an **Ewl_Tree2_View_Plain** (p. 778) to default values.*
- **Ewl_Widget * ewl_tree2_view_plain_new (void)**
*Creates a new **Ewl_Tree2_View_Plain_View**.*

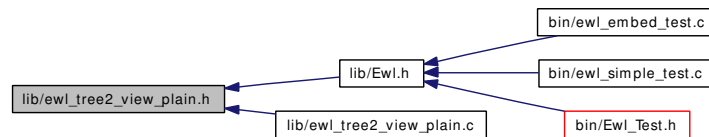
8.217 lib/ewl_tree2_view_plain.h File Reference

```
#include "ewl_tree2_view.h"
```

Include dependency graph for ewl_tree2_view_plain.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Tree2_View_Plain**

*Inherits from **Ewl_Tree2_View** (p. 776) and provides a plain tree layout.*

Defines

- `#define EWL_TREE2_VIEW_PLAIN(tv) ((Ewl_Tree2_View_Plain*)tv)`
- `#define EWL_TREE2_VIEW_PLAIN_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE2_VIEW_PLAIN_TYPE))`
- `#define EWL_TREE2_VIEW_PLAIN_TYPE "tree2_view_plain"`

Typedefs

- `typedef Ewl_Tree2_View_Plain Ewl_Tree2_View_Plain`

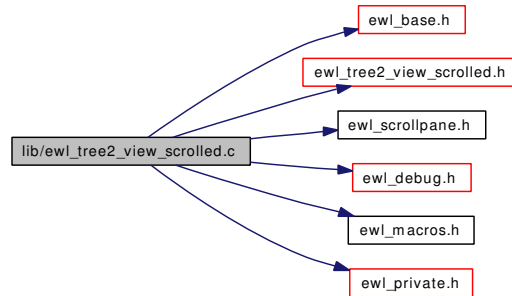
Functions

- `Ewl_View * ewl_tree2_view_plain_get (void)`
*Retrieves a shared **Ewl_Tree2_View_Plain** (p. 778) widget.*
- `int ewl_tree2_view_plain_init (Ewl_Tree2_View_Plain *tv)`
*Initializes an **Ewl_Tree2_View_Plain** (p. 778) to default values.*
- `Ewl_Widget * ewl_tree2_view_plain_new (void)`
*Creates a new **Ewl_Tree2_View_Plain_View**.*

8.218 lib/ewl_tree2_view_scrolled.c File Reference

```
#include "ewl_base.h"
#include "ewl_tree2_view_scrolled.h"
#include "ewl_scrollpane.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_tree2_view_scrolled.c:



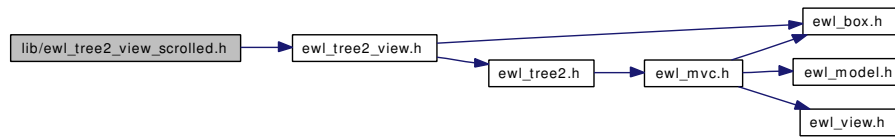
Functions

- **Ewl_View * ewl_tree2_view_scrolled_get (void)**
Retrieves the view for this widget.
- **int ewl_tree2_view_scrolled_init (Ewl_Tree2_View_Scrolled *tv)**
Initializes a tree view to default values.
- **Ewl_Widget * ewl_tree2_view_scrolled_new (void)**
*Creates and initializes a new **Ewl_Tree2_View_Scrolled** (p. 779) widget.*
- **unsigned int ewl_tree2_view_scrolled_scroll_headers_get (Ewl_Tree2_View *view)**
Retrieves if the tree headers will be scrolled or not.
- **void ewl_tree2_view_scrolled_scroll_headers_set (Ewl_Tree2_View *view, unsigned int scroll)**
Sets if the headers should be scrolled with the tree.

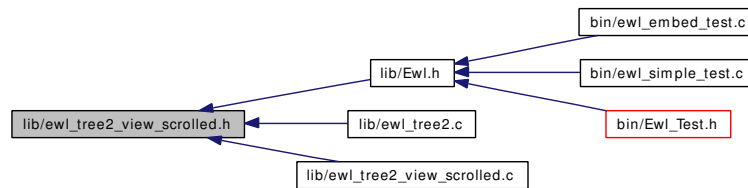
8.219 lib/ewl_tree2_view_scrolled.h File Reference

```
#include "ewl_tree2_view.h"
```

Include dependency graph for ewl_tree2_view_scrolled.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Tree2_View_Scrolled**

*Inherits from **Ewl_Box** (p. 586) and provides a scrolled tree layout.*

Defines

- `#define EWL_TREE2_VIEW_SCROLLED(tv) ((Ewl_Tree2_View_Scrolled*)tv)`
- `#define EWL_TREE2_VIEW_SCROLLED_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_TREE2_VIEW_SCROLLED_TYPE))`
- `#define EWL_TREE2_VIEW_SCROLLED_TYPE "tree2_view_scrolled"`

Typedefs

- typedef **Ewl_Tree2_View_Scrolled** **Ewl_Tree2_View_Scrolled**

Functions

- **Ewl_View *** **ewl_tree2_view_scrolled_get** (void)
Retrieves the view for this widget.
- **int** **ewl_tree2_view_scrolled_init** (**Ewl_Tree2_View_Scrolled** *tv)
Initializes a tree view to default values.
- **Ewl_Widget *** **ewl_tree2_view_scrolled_new** (void)

*Creates and initializes a new **Ewl_Tree2_View_Scrolled** (p. 779) widget.*

- unsigned int **ewl_tree2_view_scrolled_scroll_headers_get** (**Ewl_Tree2_View** *view)

Retrieves if the tree headers will be scrolled or not.

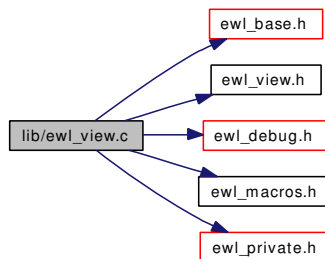
- void **ewl_tree2_view_scrolled_scroll_headers_set** (**Ewl_Tree2_View** *view, unsigned int scroll)

Sets if the headers should be scrolled with the tree.

8.220 lib/ewl_view.c File Reference

```
#include "ewl_base.h"
#include "ewl_view.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
#include "ewl_private.h"
```

Include dependency graph for ewl_view.c:



Functions

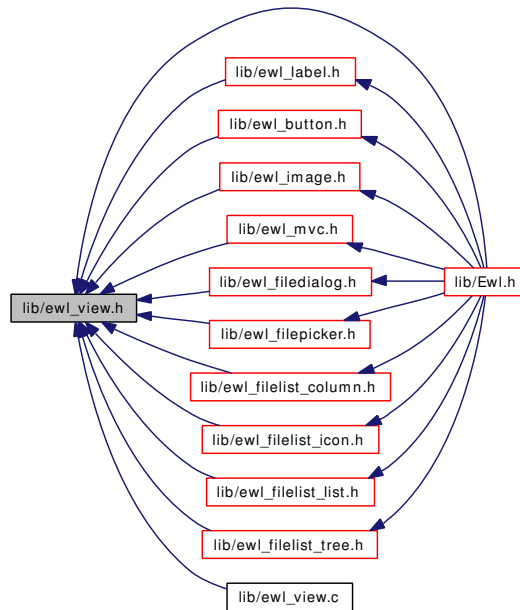
- **Ewl_View * ewl_view_clone (Ewl_View *src)**
*Creates a new **Ewl_View** (p. 784) object.*
- **Ewl_View_Expansion_View_Fetch ewl_view_expansion_view_fetch_get (Ewl_View *v)**
Gets the expansion view fetch callback from the view.
- **void ewl_view_expansion_view_fetch_set (Ewl_View *v, Ewl_View_Expansion_View_Fetch f)**
Sets the expansion view fetch callback into the view.
- **Ewl_View_Header_Fetch ewl_view_header_fetch_get (Ewl_View *v)**
Gets the header fetch callback from the view.
- **void ewl_view_header_fetch_set (Ewl_View *v, Ewl_View_Header_Fetch f)**
Sets the header fetch callback into the view.
- **int ewl_view_init (Ewl_View *view)**
*Initializes an **Ewl_View** (p. 784) object to default values.*
- **Ewl_View * ewl_view_new (void)**
*Creates a new **Ewl_View** (p. 784) object.*
- **Ewl_View_Widget_Fetch ewl_view_widget_fetch_get (Ewl_View *v)**
Get the constructor set on this view.

- void ewl_view_widget_fetch_set (Ewl_View *v, Ewl_View_Widget_Fetch fetch)

This will set the given widget fetch callback into the view.

8.221 lib/ewl_view.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_View**
The view function pointers.

Defines

- #define **EWL_VIEW**(view) ((Ewl_View *)view)
- #define **EWL_VIEW_EXPANSION_VIEW_FETCH_GET**(f) ((Ewl_View_Expansion_View_Fetch)f)
- #define **EWL_VIEW_HEADER_GET**(f) ((Ewl_View_Header_Fetch)f)
- #define **EWL_VIEW_WIDGET_FETCH**(f) ((Ewl_View_Widget_Fetch)f)

Typedefs

- typedef **Ewl_View** **Ewl_View**
- typedef **Ewl_View** (*) **Ewl_View_Expansion_View_Fetch** (void *data, int row)
- typedef **Ewl_Widget** (*) **Ewl_View_Header_Fetch** (void *data, int column)
- typedef **Ewl_Widget** (*) **Ewl_View_Widget_Fetch** (void *data, int row, int col)

Functions

- **Ewl_View** * **ewl_view_clone** (**Ewl_View** *src)

*Creates a new **Ewl_View** (p. 784) object.*

- **Ewl_View_Expansion_View_Fetch** **ewl_view_expansion_view_fetch_get** (**Ewl_View** *v)

Gets the expansion view fetch callback from the view.

- void **ewl_view_expansion_view_fetch_set** (**Ewl_View** *v, **Ewl_View_Expansion_View_Fetch** f)

Sets the expansion view fetch callback into the view.

- **Ewl_View_Header_Fetch** **ewl_view_header_fetch_get** (**Ewl_View** *v)

Gets the header fetch callback from the view.

- void **ewl_view_header_fetch_set** (**Ewl_View** *v, **Ewl_View_Header_Fetch** f)

Sets the header fetch callback into the view.

- int **ewl_view_init** (**Ewl_View** *view)

*Initializes an **Ewl_View** (p. 784) object to default values.*

- **Ewl_View** * **ewl_view_new** (void)

*Creates a new **Ewl_View** (p. 784) object.*

- **Ewl_View_Widget_Fetch** **ewl_view_widget_fetch_get** (**Ewl_View** *view)

Get the constructor set on this view.

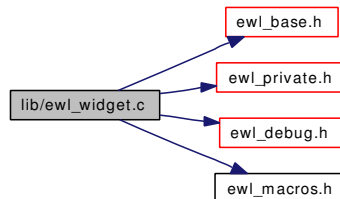
- void **ewl_view_widget_fetch_set** (**Ewl_View** *view, **Ewl_View_Widget_Fetch** construct)

This will set the given widget fetch callback into the view.

8.222 lib/ewl_widget.c File Reference

```
#include "ewl_base.h"
#include "ewl_private.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
```

Include dependency graph for ewl_widget.c:



Functions

- `char * ewl_widget_appearance_get (Ewl_Widget *w)`
Retrieve the appearance key of the widget.
- `const char * ewl_widget_appearance_part_text_get (Ewl_Widget *w, const char *part)`
Retrieve a copy of a parts current text.
- `void ewl_widget_appearance_part_text_set (Ewl_Widget *w, const char *part, const char *text)`
Change the text of the given theme part of a widget.
- `int ewl_widget_appearance_path_copy (Ewl_Widget *w, char *buf, int size)`
- `char * ewl_widget_appearance_path_get (Ewl_Widget *w)`
Retrieve the appearance path key of the widget.
- `int ewl_widget_appearance_path_size_get (Ewl_Widget *w)`
- `void ewl_widget_appearance_set (Ewl_Widget *w, const char *appearance)`
Change the appearance of the specified widget.
- `const char * ewl_widget_appearance_text_get (Ewl_Widget *w)`
Retrieve the text of the given theme part of a widget.
- `void ewl_widget_appearance_text_set (Ewl_Widget *w, const char *text)`
Change the text of the given theme part of a widget.
- `void ewl_widget_cb_configure (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)`
- `void ewl_widget_cb_disable (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)`
- `void ewl_widget_cb_drag_down (Ewl_Widget *w, void *ev_data, void *user_data __UNUSED__)`

- void **ewl_widget_cb_drag_move** (Ewl_Widget *w __UNUSED__, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_widget_cb_drag_up** (Ewl_Widget *w __UNUSED__, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_widget_cb_enable** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_widget_cb_focus_in** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_widget_cb_focus_out** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_widget_cb_hide** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_widget_cb_mouse_down** (Ewl_Widget *w, void *ev_data, void *user_data __UNUSED__)
- void **ewl_widget_cb_mouse_in** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_widget_cb_mouse_move** (Ewl_Widget *w, void *ev_data, void *user_data __UNUSED__)
- void **ewl_widget_cb_mouse_out** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_widget_cb_mouse_up** (Ewl_Widget *w, void *ev_data, void *user_data __UNUSED__)
- void **ewl_widget_cb_obscure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_widget_cb_realize** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_widget_cb_reparent** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_widget_cb_reveal** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_widget_cb_show** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_widget_cb_unrealize** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- unsigned int **ewl_widget_clipped_is** (Ewl_Widget *w)

Checks if a widget clips it's theme object.

- void **ewl_widget_clipped_set** (Ewl_Widget *w, unsigned int val)

Marks whether the widget should be clipped at it's boundaries.

- void **ewl_widget_color_get** (Ewl_Widget *w, int *r, int *g, int *b, int *a)

Gets the colour settings of the widget.

- void **ewl_widget_color_set** (Ewl_Widget *w, int r, int g, int b, int a)

sets the colour of the widget

- void **ewl_widget_configure** (Ewl_Widget *w)

Initiate configuring of the specified widget.

- void * **ewl_widget_data_del** (Ewl_Widget *w, void *k)

Remove the specified key / value pair from the widget and return the value.

- **void * ewl_widget_data_get (Ewl_Widget *w, void *k)**
retrieve the specified key / value pair from the widget
- **void ewl_widget_data_set (Ewl_Widget *w, void *k, void *v)**
Attach the specified key / value pair to the widget.
- **void ewl_widget_destroy (Ewl_Widget *w)**
Destroy the specified widget.
- **void ewl_widget_disable (Ewl_Widget *w)**
Prevent a widget from receiving any events.
- **void ewl_widget_dnd_reset (void)**
Cancel any active dnd_wait state widgets.
- **Ewl_Widget * ewl_widget_drag_candidate_get (void)**
Accessor function for the current drag candidate widget.
- **void ewl_widget_draggable_set (Ewl_Widget *w, unsigned int val, Ewl_Widget_Drag cb)**
Set the draggable state, and setup any callbacks.
- **void ewl_widget_enable (Ewl_Widget *w)**
Re-enable a disabled widget.
- **void ewl_widget_focus_send (Ewl_Widget *w)**
Changes the keyboard focus to the widget w.
- **unsigned int ewl_widget_focusable_get (Ewl_Widget *w)**
Checks the focusable state of the widget.
- **void ewl_widget_focusable_set (Ewl_Widget *w, unsigned int val)**
Set if the given widget is focusable or not.
- **Ewl_Widget * ewl_widget_focused_get (void)**
Retrieve the currently focused widget.
- **void ewl_widget_free (Ewl_Widget *w)**
- **void ewl_widget_hide (Ewl_Widget *w)**
Mark a widget as invisible.
- **unsigned int ewl_widget_ignore_focus_change_get (Ewl_Widget *w)**
Get if the widget is ignoring focus changes.
- **void ewl_widget_ignore_focus_change_set (Ewl_Widget *w, unsigned int val)**
Set if the widget should ignore focus changes.
- **void ewl_widget_inherit (Ewl_Widget *widget, const char *inherit)**
Appends the given inheritance to this widgets inheritance string.

- **int ewl_widget_init (Ewl_Widget *w)**
Initialize a widget to default values and callbacks.
- **unsigned int ewl_widget_internal_is (Ewl_Widget *w)**
- **void ewl_widget_internal_set (Ewl_Widget *w, unsigned int val)**
- **int ewl_widget_layer_priority_get (Ewl_Widget *w)**
Retrieve a widgets layer relative to it's parent.
- **void ewl_widget_layer_priority_set (Ewl_Widget *w, int layer)**
Set the relative layer to it's parent.
- **int ewl_widget_layer_top_get (Ewl_Widget *w)**
Returns if the widget will be drawn above all the others.
- **void ewl_widget_layer_top_set (Ewl_Widget *w, int top)**
set the widget to be layered above all other widgets
- **Ewl_Widget * ewl_widget_name_find (const char *name)**
Find a widget identified by a name.
- **const char * ewl_widget_name_get (Ewl_Widget *w)**
Get the name for the specified widget.
- **void ewl_widget_name_set (Ewl_Widget *w, const char *name)**
Name the specified widget.
- **Ewl_Widget * ewl_widget_new (void)**
Allocate a new widget.
- **void ewl_widget_obscure (Ewl_Widget *w)**
Indicate a widget is obscured.
- **unsigned int ewl_widget_onscreen_is (Ewl_Widget *w)**
Checks if the given widget is currently on screen.
- **Ewl_Widget * ewl_widget_parent_get (Ewl_Widget *w)**
Retrieves the parent of the given widget.
- **int ewl_widget_parent_of (Ewl_Widget *c, Ewl_Widget *w)**
Determine if a widget is a parent of another widget.
- **void ewl_widget_parent_set (Ewl_Widget *w, Ewl_Widget *p)**
change the parent of the specified widget
- **void ewl_widget_print (Ewl_Widget *w)**
Prints info for debugging a widget's state information.
- **void ewl_widget_print_verbose (Ewl_Widget *w)**
Prints verbose info for debugging a widget's state information.

- void **ewl_widget_realize** (**Ewl_Widget** *w)
Realize the specified widget.
- void **ewl_widget_reparent** (**Ewl_Widget** *w)
initiate reparent of the specified widget
- void **ewl_widget_reveal** (**Ewl_Widget** *w)
Indicate a widget is revealed.
- void **ewl_widget_show** (**Ewl_Widget** *w)
mark a widget as visible
- void **ewl_widget_state_set** (**Ewl_Widget** *w, const char *state, **Ewl_State_Type** flag)
Update the appearance of the widget to a state.
- void **ewl_widget_tab_order_append** (**Ewl_Widget** *w)
Changes the order in the embed so w receives focus first on tab.
- void **ewl_widget_tab_order_insert** (**Ewl_Widget** *w, unsigned int idx)
Changes the order in the embed so w receives focus first on tab.
- void **ewl_widget_tab_order_insert_after** (**Ewl_Widget** *w, **Ewl_Widget** *after)
Insert the given widget into the tab order after the after widget.
- void **ewl_widget_tab_order_insert_before** (**Ewl_Widget** *w, **Ewl_Widget** *before)
Inserts the widget into the tab order before the before widget.
- void **ewl_widget_tab_order_prepend** (**Ewl_Widget** *w)
Changes the order in the embed so w receives focus first on tab.
- void **ewl_widget_tab_order_remove** (**Ewl_Widget** *w)
Remove the widget from the tab order.
- void **ewl_widget_tree_print** (**Ewl_Widget** *w)
Prints to stdout the tree of widgets that are parents of a widget.
- unsigned int **ewl_widget_type_is** (**Ewl_Widget** *widget, const char *type)
Determine if the widget w has inherited from the type t.
- void **ewl_widget_unrealize** (**Ewl_Widget** *w)
Unrealize the specified widget.

8.222.1 Function Documentation

- 8.222.1.1 void ewl_widget_cb_configure (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
 - 8.222.1.2 void ewl_widget_cb_disable (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
 - 8.222.1.3 void ewl_widget_cb_drag_down (Ewl_Widget * *w*, void * *ev_data*,
void *user_data __ *UNUSED* __)
 - 8.222.1.4 void ewl_widget_cb_drag_move (Ewl_Widget * *w* __ *UNUSED* __,
void *ev_data __ *UNUSED* __, void *user_data __ *UNUSED* __)
 - 8.222.1.5 void ewl_widget_cb_drag_up (Ewl_Widget * *w* __ *UNUSED* __, void
*ev_data __ *UNUSED* __, void *user_data __ *UNUSED* __)
 - 8.222.1.6 void ewl_widget_cb_enable (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
 - 8.222.1.7 void ewl_widget_cb_focus_in (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
 - 8.222.1.8 void ewl_widget_cb_focus_out (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
 - 8.222.1.9 void ewl_widget_cb_hide (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
 - 8.222.1.10 void ewl_widget_cb_mouse_down (Ewl_Widget * *w*, void * *ev_data*,
void *user_data __ *UNUSED* __)
 - 8.222.1.11 void ewl_widget_cb_mouse_in (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
 - 8.222.1.12 void ewl_widget_cb_mouse_move (Ewl_Widget * *w*, void * *ev_data*,
void *user_data __ *UNUSED* __)
 - 8.222.1.13 void ewl_widget_cb_mouse_out (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
 - 8.222.1.14 void ewl_widget_cb_mouse_up (Ewl_Widget * *w*, void * *ev_data*,
void *user_data __ *UNUSED* __)
 - 8.222.1.15 void ewl_widget_cb_obscure (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
 - 8.222.1.16 void ewl_widget_cb_realize (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
 - 8.222.1.17 void ewl_widget_cb_reparent (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
 - 8.222.1.18 void ewl_widget_cb_reveal (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
-
- Generated on Wed Mar 28 00:01:51 2007 for Enlightened Widget Library by Doxygen
- 8.222.1.19 void ewl_widget_cb_show (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
 - 8.222.1.20 void ewl_widget_cb_unrealize (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)

Returns:

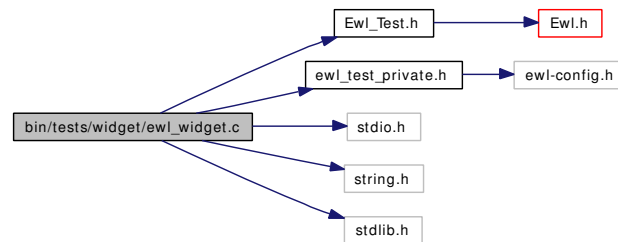
Returns a newly allocated widget on success, NULL on failure.

Do not use this function unless you know what you are doing! It is only intended to easily create custom widgets that are not containers.

8.223 bin/tests/widget/ewl_widget.c File Reference

```
#include "Ewl_Test.h"
#include "ewl_test_private.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
```

Include dependency graph for ewl_widget.c:



Functions

- `void test_info (Ewl_Test *test)`

8.223.1 Function Documentation

8.223.1.1 `void test_info (Ewl_Test * test)`

8.224 lib/ewl_widget.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Attach_List**
A list of things attached to a widget.
- struct **Ewl_Callback_Chain**
The callback chain contains the length, mask and information on the list.
- struct **Ewl_Color_Set**
Contains an RGBA set of colours.
- struct **Ewl_Pair**
Contains a key and a value pair.
- struct **Ewl_Pair_List**
Contains a list of key value pairs.
- struct **Ewl_Widget**
*Inherits from **Ewl_Object** (p. 709) and extends to provide appearance, parent, and callback capabilities.*

Defines

- `#define EWL_PAIR(p) ((Ewl_Pair *)p)`
- `#define EWL_WIDGET(widget) ((Ewl_Widget *) widget)`
*Typecast a pointer to an **Ewl_Widget** (p. 785) pointer.*
- `#define EWL_WIDGET_IS(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_WIDGET_TYPE))`
- `#define EWL_WIDGET_TYPE "widget"`

Typedefs

- `typedef Ewl_Attach_List Ewl_Attach_List`
- `typedef Ewl_Callback_Chain Ewl_Callback_Chain`
- `typedef Ewl_Color_Set Ewl_Color_Set`
- `typedef Ewl_Pair Ewl_Pair`
- `typedef Ewl_Pair_List Ewl_Pair_List`
- `typedef Ewl_Widget Ewl_Widget`
- `typedef void (*)(Ewl_Widget_Drag) (void)`

Functions

- `char * ewl_widget_appearance_get (Ewl_Widget *w)`
Retrieve the appearance key of the widget.
- `const char * ewl_widget_appearance_part_text_get (Ewl_Widget *w, const char *part)`
Retrieve a copy of a parts current text.
- `void ewl_widget_appearance_part_text_set (Ewl_Widget *w, const char *part, const char *text)`
Change the text of the given theme part of a widget.
- `int ewl_widget_appearance_path_copy (Ewl_Widget *w, char *buf, int size)`
- `char * ewl_widget_appearance_path_get (Ewl_Widget *w)`
Retrieve the appearance path key of the widget.
- `int ewl_widget_appearance_path_size_get (Ewl_Widget *w)`
- `void ewl_widget_appearance_set (Ewl_Widget *w, const char *appearance)`
Change the appearance of the specified widget.
- `const char * ewl_widget_appearance_text_get (Ewl_Widget *w)`
Retrieve the text of the given theme part of a widget.
- `void ewl_widget_appearance_text_set (Ewl_Widget *w, const char *text)`
Change the text of the given theme part of a widget.
- `void ewl_widget_cb_configure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_disable (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_enable (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_focus_in (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_focus_out (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_hide (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_mouse_down (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_mouse_in (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_mouse_move (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_mouse_out (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_mouse_up (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_obscure (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_realize (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_reparent (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_reveal (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_show (Ewl_Widget *w, void *ev_data, void *user_data)`
- `void ewl_widget_cb_unrealize (Ewl_Widget *w, void *ev_data, void *user_data)`
- `unsigned int ewl_widget_clipped_is (Ewl_Widget *w)`
Checks if a widget clips it's theme object.
- `void ewl_widget_clipped_set (Ewl_Widget *w, unsigned int val)`

Marks whether the widget should be clipped at it's boundaries.

- void **ewl_widget_color_get** (**Ewl_Widget** *w, int *r, int *g, int *b, int *a)
Gets the colour settings of the widget.
- void **ewl_widget_color_set** (**Ewl_Widget** *w, int r, int g, int b, int a)
sets the colour of the widget
- void **ewl_widget_configure** (**Ewl_Widget** *widget)
Initiate configuring of the specified widget.
- void * **ewl_widget_data_del** (**Ewl_Widget** *w, void *k)
Remove the specified key / value pair from the widget and return the value.
- void * **ewl_widget_data_get** (**Ewl_Widget** *w, void *k)
retrieve the specified key / value pair from the widget
- void **ewl_widget_data_set** (**Ewl_Widget** *w, void *k, void *v)
Attach the specified key / value pair to the widget.
- void **ewl_widget_destroy** (**Ewl_Widget** *widget)
Destroy the specified widget.
- void **ewl_widget_disable** (**Ewl_Widget** *w)
Prevent a widget from receiving any events.
- void **ewl_widget_dnd_reset** (void)
Cancel any active dnd_wait state widgets.
- **Ewl_Widget** * **ewl_widget_drag_candidate_get** (void)
Accessor function for the current drag candidate widget.
- void **ewl_widget_draggable_set** (**Ewl_Widget** *w, unsigned int val, **Ewl_Widget_Drag** cb)
Set the draggable state, and setup any callbacks.
- void **ewl_widget_enable** (**Ewl_Widget** *w)
Re-enable a disabled widget.
- void **ewl_widget_focus_send** (**Ewl_Widget** *w)
Changes the keyboard focus to the widget w.
- unsigned int **ewl_widget_focusable_get** (**Ewl_Widget** *w)
Checks the focusable state of the widget.
- void **ewl_widget_focusable_set** (**Ewl_Widget** *w, unsigned int val)
Set if the given widget is focusable or not.
- **Ewl_Widget** * **ewl_widget_focused_get** (void)
Retrieve the currently focused widget.

- void **ewl_widget_free** (Ewl_Widget *w)
- void **ewl_widget_hide** (Ewl_Widget *widget)
Mark a widget as invisible.
- unsigned int **ewl_widget_ignore_focus_change_get** (Ewl_Widget *w)
Get if the widget is ignoring focus changes.
- void **ewl_widget_ignore_focus_change_set** (Ewl_Widget *w, unsigned int val)
Set if the widget should ignore focus changes.
- void **ewl_widget_inherit** (Ewl_Widget *widget, const char *type)
Appends the given inheritance to this widgets inheritance string.
- int **ewl_widget_init** (Ewl_Widget *w)
Initialize a widget to default values and callbacks.
- unsigned int **ewl_widget_internal_is** (Ewl_Widget *w)
- void **ewl_widget_internal_set** (Ewl_Widget *w, unsigned int val)
- int **ewl_widget_layer_priority_get** (Ewl_Widget *w)
Retrieve a widgets layer relative to it's parent.
- void **ewl_widget_layer_priority_set** (Ewl_Widget *w, int layer)
Set the relative layer to it's parent.
- int **ewl_widget_layer_top_get** (Ewl_Widget *w)
Returns if the widget will be drawn above all the others.
- void **ewl_widget_layer_top_set** (Ewl_Widget *w, int top)
set the widget to be layered above all other widgets
- Ewl_Widget * **ewl_widget_name_find** (const char *name)
Find a widget identified by a name.
- const char * **ewl_widget_name_get** (Ewl_Widget *w)
Get the name for the specified widget.
- void **ewl_widget_name_set** (Ewl_Widget *w, const char *name)
Name the specified widget.
- void **ewl_widget_obscure** (Ewl_Widget *w)
Indicate a widget is obscured.
- unsigned int **ewl_widget_onscreen_is** (Ewl_Widget *widget)
Checks if the given widget is currently on screen.
- Ewl_Widget * **ewl_widget_parent_get** (Ewl_Widget *w)
Retrieves the parent of the given widget.
- int **ewl_widget_parent_of** (Ewl_Widget *c, Ewl_Widget *w)

Determine if a widget is a parent of another widget.

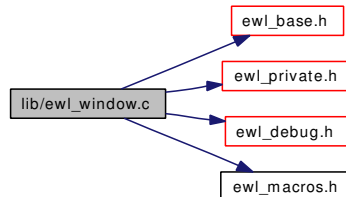
- void **ewl_widget_parent_set** (**Ewl_Widget** *w, **Ewl_Widget** *p)
change the parent of the specified widget
- void **ewl_widget_print** (**Ewl_Widget** *w)
Prints info for debugging a widget's state information.
- void **ewl_widget_print_verbose** (**Ewl_Widget** *w)
Prints verbose info for debugging a widget's state information.
- void **ewl_widget_realize** (**Ewl_Widget** *widget)
Realize the specified widget.
- void **ewl_widget_reparent** (**Ewl_Widget** *widget)
initiate reparent of the specified widget
- void **ewl_widget_reveal** (**Ewl_Widget** *w)
Indicate a widget is revealed.
- void **ewl_widget_show** (**Ewl_Widget** *widget)
mark a widget as visible
- void **ewl_widget_state_set** (**Ewl_Widget** *w, const char *state, **Ewl_State_Type** flag)
Update the appearance of the widget to a state.
- void **ewl_widget_tab_order_append** (**Ewl_Widget** *w)
Changes the order in the embed so w receives focus first on tab.
- void **ewl_widget_tab_order_insert** (**Ewl_Widget** *w, unsigned int idx)
Changes the order in the embed so w receives focus first on tab.
- void **ewl_widget_tab_order_insert_after** (**Ewl_Widget** *w, **Ewl_Widget** *after)
Insert the given widget into the tab order after the after widget.
- void **ewl_widget_tab_order_insert_before** (**Ewl_Widget** *w, **Ewl_Widget** *before)
Inserts the widget into the tab order before the before widget.
- void **ewl_widget_tab_order_prepend** (**Ewl_Widget** *w)
Changes the order in the embed so w receives focus first on tab.
- void **ewl_widget_tab_order_remove** (**Ewl_Widget** *w)
Remove the widget from the tab order.
- void **ewl_widget_tree_print** (**Ewl_Widget** *w)
Prints to stdout the tree of widgets that are parents of a widget.

- unsigned int **ewl_widget_type_is** (**Ewl_Widget** *widget, const char *type)
Determine if the widget w has inherited from the type t.
- void **ewl_widget_unrealize** (**Ewl_Widget** *w)
Unrealize the specified widget.

8.225 lib/ewl_window.c File Reference

```
#include "ewl_base.h"
#include "ewl_private.h"
#include "ewl_debug.h"
#include "ewl_macros.h"
```

Include dependency graph for ewl_window.c:



Functions

- void **ewl_window_attention_demand** (Ewl_Window *win)
Request the WM to pay attention to the window.
- void **ewl_window_borderless_set** (Ewl_Window *win)
Remove the border from the specified window.
- void **ewl_window_cb_configure** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_window_cb_destroy** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_window_cb_expose** (Ewl_Widget *w, void *ev __UNUSED__, void *user_data __UNUSED__)
- void **ewl_window_cb_hide** (Ewl_Widget *widget, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_window_cb_postrealize** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_window_cb_realize** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_window_cb_realize_transient** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data)
- void **ewl_window_cb_show** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- void **ewl_window_cb_unrealize** (Ewl_Widget *w, void *ev_data __UNUSED__, void *user_data __UNUSED__)
- char * **ewl_window_class_get** (Ewl_Window *win)
Retrieve the class of the specified window.
- void **ewl_window_class_set** (Ewl_Window *win, const char *classname)
Set the class of the specified window.
- int **ewl_window_dialog_get** (Ewl_Window *win)

Retrieves the current dialog state on a window.

- void **ewl_window_dialog_set** (**Ewl_Window** *win, int dialog)
Changes the dialog state on the specified window.
- unsigned int **ewl_window_fullscreen_get** (**Ewl_Window** *win)
Retrieve the fullscreen setting for the window.
- void **ewl_window_fullscreen_set** (**Ewl_Window** *win, unsigned int fullscreen)
Sets the fullscreen setting for the window.
- int **ewl_window_init** (**Ewl_Window** *w)
Initialize a window to default values and callbacks.
- int **ewl_window_keyboard_grab_get** (**Ewl_Window** *win)
Retrieves the current keyboard grab state on a window.
- void **ewl_window_keyboard_grab_set** (**Ewl_Window** *win, int grab)
Changes the keyboard grab state on the specified window.
- void **ewl_window_lower** (**Ewl_Window** *win)
Lower a window.
- int **ewl_window_modal_get** (**Ewl_Window** *win)
Gets the boolean flag indicating if win is modal.
- void **ewl_window_modal_set** (**Ewl_Window** *win, int modal)
Sets the window to modal or non-modal based on modal.
- void **ewl_window_move** (**Ewl_Window** *win, int x, int y)
Move the specified window to the given position.
- char * **ewl_window_name_get** (**Ewl_Window** *win)
Retrieve the name of the specified window.
- void **ewl_window_name_set** (**Ewl_Window** *win, const char *name)
Set the name of the specified window.
- **Ewl_Widget** * **ewl_window_new** (void)
Allocate and initialize a new window.
- int **ewl_window_override_get** (**Ewl_Window** *win)
Retrieves the current override state on a window.
- void **ewl_window_override_set** (**Ewl_Window** *win, int override)
Changes the override state on the specified window.
- int **ewl_window_pointer_grab_get** (**Ewl_Window** *win)
Retrieves the current pointer grab state on a window.

- void **ewl_window_pointer_grab_set** (**Ewl_Window** *win, int grab)
Changes the pointer grab state on the specified window.
- void **ewl_window_raise** (**Ewl_Window** *win)
Raise a window.
- void **ewl_window_selection_text_set** (**Ewl_Window** *win, const char *txt)
This will set the given txt as the selection text on the window or clear the text if txt is NULL.
- unsigned int **ewl_window_skip_pager_get** (**Ewl_Window** *win)
Retrieve the skip pager setting for the window.
- void **ewl_window_skip_pager_set** (**Ewl_Window** *win, unsigned int skip)
Sets the skip pager setting for the window.
- unsigned int **ewl_window_skip_taskbar_get** (**Ewl_Window** *win)
Retrieve the skip taskbar setting for the window.
- void **ewl_window_skip_taskbar_set** (**Ewl_Window** *win, unsigned int skip)
Sets the skip taskbar setting for the window.
- char * **ewl_window_title_get** (**Ewl_Window** *win)
Retrieve the title of the specified window.
- void **ewl_window_title_set** (**Ewl_Window** *win, const char *title)
Set the title of the specified window.
- void **ewl_window_transient_for** (**Ewl_Window** *win, **Ewl_Window** *forwin)
Sets a window to be transient for another window.
- void **ewl_window_transient_for_foreign** (**Ewl_Window** *win, **Ewl_Embed_Window** *forwin)
Sets a window to be transient for another window.
- **Ewl_Window** * **ewl_window_window_find** (void *window)
Find an ewl window by it's X window.

Variables

- int **EWL_CALLBACK_DELETE_WINDOW** = 0
- int **EWL_CALLBACK_EXPOSE** = 0
- **Ecore_List** * **ewl_window_list** = NULL

8.225.1 Function Documentation

- 8.225.1.1 void ewl_window_cb_configure (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.225.1.2 void ewl_window_cb_destroy (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.225.1.3 void ewl_window_cb_expose (Ewl_Widget * *w*, void *ev
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.225.1.4 void ewl_window_cb_hide (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.225.1.5 void ewl_window_cb_postrealize (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.225.1.6 void ewl_window_cb_realize (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.225.1.7 void ewl_window_cb_realize_transient (Ewl_Widget * *w*, void
*ev_data __ *UNUSED* __, void * *user_data*)
- 8.225.1.8 void ewl_window_cb_show (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)
- 8.225.1.9 void ewl_window_cb_unrealize (Ewl_Widget * *w*, void *ev_data
__ *UNUSED* __, void *user_data __ *UNUSED* __)

8.225.2 Variable Documentation

- 8.225.2.1 Ecore_List* ewl_window_list = NULL

8.226 lib/ewl_window.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct **Ewl_Window**

*Inherits from **Ewl_Embed** (p. 629) class to create it's own window and canvas for drawing, sizing and positioning.*

Defines

- #define **EWL_WINDOW**(win) ((**Ewl_Window** *) win)
- #define **EWL_WINDOW_IS**(w) (ewl_widget_type_is(EWL_WIDGET(w), EWL_WINDOW_TYPE))
- #define **EWL_WINDOW_TYPE** "window"

Typedefs

- typedef **Ewl_Window** **Ewl_Window**

Functions

- void **ewl_window_attention_demand** (**Ewl_Window** *win)
Request the WM to pay attention to the window.
- void **ewl_window_borderless_set** (**Ewl_Window** *win)
Remove the border from the specified window.
- void **ewl_window_cb_configure** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_window_cb_destroy** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_window_cb_expose** (**Ewl_Widget** *w, void *ev, void *user_data)
- void **ewl_window_cb_hide** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_window_cb_postrealize** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_window_cb_realize** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_window_cb_realize_transient** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_window_cb_show** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- void **ewl_window_cb_unrealize** (**Ewl_Widget** *w, void *ev_data, void *user_data)
- char * **ewl_window_class_get** (**Ewl_Window** *win)
Retrieve the class of the specified window.
- void **ewl_window_class_set** (**Ewl_Window** *win, const char *classname)
Set the class of the specified window.

- **int ewl_window_dialog_get (Ewl_Window *win)**
Retrieves the current dialog state on a window.
- **void ewl_window_dialog_set (Ewl_Window *win, int dialog)**
Changes the dialog state on the specified window.
- **unsigned int ewl_window_fullscreen_get (Ewl_Window *win)**
Retrieve the fullscreen setting for the window.
- **void ewl_window_fullscreen_set (Ewl_Window *win, unsigned int fullscreen)**
Sets the fullscreen setting for the window.
- **int ewl_window_init (Ewl_Window *win)**
Initialize a window to default values and callbacks.
- **int ewl_window_keyboard_grab_get (Ewl_Window *win)**
Retrieves the current keyboard grab state on a window.
- **void ewl_window_keyboard_grab_set (Ewl_Window *win, int grab)**
Changes the keyboard grab state on the specified window.
- **void ewl_window_lower (Ewl_Window *win)**
Lower a window.
- **int ewl_window_modal_get (Ewl_Window *win)**
Gets the boolean flag indicating if win is modal.
- **void ewl_window_modal_set (Ewl_Window *win, int modal)**
Sets the window to modal or non-modal based on modal.
- **void ewl_window_move (Ewl_Window *win, int x, int y)**
Move the specified window to the given position.
- **char * ewl_window_name_get (Ewl_Window *win)**
Retrieve the name of the specified window.
- **void ewl_window_name_set (Ewl_Window *win, const char *name)**
Set the name of the specified window.
- **Ewl_Widget * ewl_window_new (void)**
Allocate and initialize a new window.
- **int ewl_window_override_get (Ewl_Window *win)**
Retrieves the current override state on a window.
- **void ewl_window_override_set (Ewl_Window *win, int override)**
Changes the override state on the specified window.
- **int ewl_window_pointer_grab_get (Ewl_Window *win)**

Retrieves the current pointer grab state on a window.

- void **ewl_window_pointer_grab_set** (**Ewl_Window** *win, int grab)
Changes the pointer grab state on the specified window.
- void **ewl_window_raise** (**Ewl_Window** *win)
Raise a window.
- void **ewl_window_selection_text_set** (**Ewl_Window** *win, const char *txt)
This will set the given txt as the selection text on the window or clear the text if txt is NULL.
- unsigned int **ewl_window_skip_pager_get** (**Ewl_Window** *win)
Retrieve the skip pager setting for the window.
- void **ewl_window_skip_pager_set** (**Ewl_Window** *win, unsigned int skip)
Sets the skip pager setting for the window.
- unsigned int **ewl_window_skip_taskbar_get** (**Ewl_Window** *win)
Retrieve the skip taskbar setting for the window.
- void **ewl_window_skip_taskbar_set** (**Ewl_Window** *win, unsigned int skip)
Sets the skip taskbar setting for the window.
- char * **ewl_window_title_get** (**Ewl_Window** *win)
Retrieve the title of the specified window.
- void **ewl_window_title_set** (**Ewl_Window** *win, const char *title)
Set the title of the specified window.
- void **ewl_window_transient_for** (**Ewl_Window** *win, **Ewl_Window** *forwin)
Sets a window to be transient for another window.
- void **ewl_window_transient_for_foreign** (**Ewl_Window** *win, **Ewl_Embed_Window** *forwin)
Sets a window to be transient for another window.
- **Ewl_Window** * **ewl_window_window_find** (void *window)
Find an ewl window by it's X window.

Variables

- int **EWL_CALLBACK_DELETE_WINDOW**
- int **EWL_CALLBACK_EXPOSE**

8.227 /home/ewww/e17/libs/ewl/TODO File Reference

Chapter 9

Enlightened Widget Library Page Documentation

9.1 Layering Scheme

As widgets are placed inside containers, there becomes the issue of specifying which widget gets drawn on top. It's important that the widgets placed inside of a container are above the containers background, or the user would be unable to view the placed widgets.

EWL handles the layering so the programmer doesn't need to worry about it. In some cases it is necessary for a widget to overlap another widget with the same parent. In this case you set the layer priority of the widget to a higher value (negative values are possible).

If you want a widget to be placed over top of all other widgets, like floater for example, you can set the widget to be 'top layered'. This setting will also effect the children of that widget.

9.2 Documentation Images

Diagram describing **Ewl_Object** (p. 709) fields

Diagram describing how **Ewl_Object** (p. 709) fields affect sizing.

9.3 How Themes Work

EWL relies on a graphical layout engine called Edje. Edje allows a themer to layout images, in fairly advanced ways and provides methods for animation and changing of images based on various signals. EWL uses an Edje collection per-widget and has a set of keys for specifying which widget uses a specific collection. Information about using Edje can be found at the [project page](#). The remainder of this section concentrates specifically on how EWL interacts with Edje.

The keys to lookup the group name for the widgets are top level data keys in the themes edje file. Each widget looks up it's group name based on the heirarchy it is placed inside other containers.

9.4 TODO and Bugs list

9.4.1 BUGS

- We auto add things to the tab order as they are realized. We need to make sure the things we are adding aren't already in the tab order as we will then remove them. Should possibly add a flag to deactivate this behaviour if the user wants complete control over their tab order.
- When we receive a mouse down on a widget we adjust the current position in the tab order to be the clicked widget (or the first non internal widget up the tree). This needs to be changed to check both `!internal` & `lin_tab_order` as internal widgets `_can_` be added to the tab order it just isn't done by default
- The realize code will current add all non internal widgets into the tab order automatically. This needs to be updated to check that the widget is not current in the tab order already. Currently if the user sets up a specific tab order it will be whiped out by the realize code. (In **Ewl_Embed** (p. 629) I believe).

Ewl_Entry (p. 635)

- text scrolling with **Ewl_Entry** (p. 635) needs a bit of work. If you scroll over to the right and the text moves and arrow back it shifts as you move off of the right edge instead of over the left edge.
- text cursor disappears on the end of entrybox case.
- if you're in a multiline text and on a line that is longer then then one above or below and you hit the up arrow it will position your cursor one character `_before_` the end of the line.
- if you're in a multiline text and on the last line of multiple lines pressing arrow down can wrap to the beginning of the first line.

Ewl_Filepicker (p. 668)

- The column view needs to blank out the preview when you have multiple files selected.

Ewl_Freebox (p. 673)

- On the comparator test if you add a set of icons, rename an icon and then add another set of icons the second set of icons won't appear until the window is moved. (Must be missing a configure event)
- The autolayout will align at the top, should probably be changed to align at the bottom based on the largest widget being put in.

Ewl_Icon (p. 683)

- Clicking on the label 'Draw (Editable)' then clicking again will cause a warning to print. I `_think_` the move event is getting fired for the text widget after I've deleted it in the **Ewl_Icon** (p. 683) code. Need to confirm/fix this.

- Menus aren't hooked up. This should popup on right click and hide on a click outside the menu. Needs to do all that grabby stuff menus do (which should be in the menu's already).
- Should change label editing to double click on the icon instead of clicking on the label itself.

Ewl_Menu (p. 696)

- Menus will activate on tab, but it jumps to the submenu instead of each menu item

Ewl_Notebook (p. 707)

- Putting a second notebook inside a first can cause memory corruption. This can be seen by running **Ewl_Test** (p. 755) and then clicking advanced -> Theme. The tab doesn't change and using other tests causes EWL warnings to be printed out.

Ewl_Password (p. 719)

- The entry needs to correctly deal with selections and mouse clicks.

Ewl_Scrollpane (p. 732)

- How to deal with the scrollpane with the focus? Currently widgets marked INTERNAL are ignored on the initial add. Should the scrollpane find itself in the tab list on SHOW and substitute the seekers instead?

Ewl_Spectrum (p. 743)

- Add a theme item for the cross hairs ...

Ewl_Test (p. 755)

- Add a unit test to create a buffer engine window, show it, unrealize and realize it. (This currently causes a segv in the buffer engine code.)

Ewl_Text (p. 757)

- need to setup the styles/align/wrap data from the theme in `ewl_text_context_default_create`
- new theme keys for the align/wrap stuff

9.4.2 TODO and Bugs list

- Enabled/Disabled status for ALL applicable widgets.

Ewl_Colorpicker (p. 603)

- Need a slider for choosing alpha too. Need to add `_get/_set` functions to set the alpha parameter that are separate from the colour `_set/_get` functions which will just be RGB/HSV functions.

Ewl_Engines

- For engines that only support a single window (framebuffer for example) we need to set it up to override the window creation call to make the window inside an Evas buffer engine.

Ewl_Filedialog (p. 656)

- Need to add the ability to select directories instead of just files

Ewl_Radiobutton (p. 725)

- Add a function that returns the selected radio button of a chain associated to a given radio button.

Ewl_Seeker (p. 734)

- Add value labels and hash marks.

Ewl_Spectrum (p. 743)

- Implement a triangle selection mode

Ewl_Statusbar (p. 748)

- Need to return a context when items are pushed onto the status bar so that a context can be removed without popping off the stack

Ewl_Test (p. 755)

- More Tutorials in the src/bin/test/*.c files
- More unit tests. There are a few examples in there but we need to add unit tests for as many functions as we can.

Ewl_Widget (p. 785)

- Use a flag instead of the variable **Ewl_Widget::toplayered** (p. 787)

9.5 Widget Theme Keys

Global EWL_BORDER_TYPE (p. 27) /border/file
/border/group

Global EWL_BOX_TYPE (p. 31) /box/file
/box/group

Global EWL_BUTTON_TYPE (p. 36) /button/file
/button/group

Global EWL_CALENDAR_TYPE (p. 42) /calendar/file
/calendar/group

Global EWL_CELL_TYPE (p. 52) /cell/file
/cell/group

Global EWL_CHECK_TYPE (p. 54) /check/file
/check/group

Global EWL_CHECKBUTTON_TYPE (p. 57) /checkboxbutton/file
/checkboxbutton/group

Global EWL_COMBO_TYPE (p. 73) /combo/file
/combo/group

Global EWL_CURSOR_TYPE (p. 105) /cursor/file
/cursor/group

Global EWL_DATEPICKER_TYPE (p. 107) /datepicker/file
/datepicker/group

Global EWL_DIALOG_TYPE (p. 109) /dialog/file
/dialog/group

Global EWL_EMBED_TYPE (p. 122) /embed/file
/embed/group

Global EWL_EXPANSION_TYPE (p. 171) /expansion/group

Global EWL_FILEDIALOG_TYPE (p. 175) /filedialog/file
/filedialog/group

Global EWL_FLOATER_TYPE (p. 211) /floater/file
/floater/group

Global EWL_GRID_TYPE (p. 220) /grid/file
/grid/group

Global EWL_IMAGE_TYPE (p. 279) /image/file
/image/group

Global EWL_LABEL_TYPE (p. 288) /label

Global EWL_MEDIA_TYPE (p. 297) /media/file
/media/group

Global EWL_MENU_TYPE (p. 303) /menu/file
/menu/group

Global EWL_MENU_ITEM_TYPE (p. 307) /menu_item/file
/menu_item/group

Global EWL_MENUBAR_TYPE (p. 309) /menubar/file
/menubar/group

Global EWL_NOTEBOOK_TYPE (p. 337) /notebook
/notebook/tabbar
/notebook/tabbar/
/notebook/tabbar/tab
/notebook/tabbar/tab/label
/notebook/pages

Global EWL_OVERLAY_TYPE (p. 373) /overlay/file
/overlay/group

Global EWL_PANED_TYPE (p. 376) /paned/file

/paned/group
/grabber/vertical/file
/grabber/vertical/group
/grabber/horizontal/file
/grabber/horizontal/group

Global EWL_PASSWORD_TYPE (p. 380) /password/file

/password/group

Global EWL_PROGRESSBAR_TYPE (p. 389) /progressbar/file

/progressbar/group
/bar/file
/bar/group

Global EWL_RADIOBUTTON_TYPE (p. 393) /radiobutton/file

/radiobutton/group

Global EWL_ROW_TYPE (p. 402) /row/file

/row/group

Global EWL_SCROLLBAR_TYPE (p. 405) /hscrollbar/file

/hscrollbar/group
/vscrollbar/file
/vscrollbar/group
/hscrollbar/seeker/group
/vscrollbar/seeker/file
/vscrollbar/seeker/group
/hscrollbar/seeker/button/group
/vscrollbar/seeker/button/file
/vscrollbar/seeker/button/group

Global EWL_SCROLLBAR_TYPE (p. 405) /hscrollbar/seeker/file**Global EWL_SCROLLBAR_TYPE** (p. 405) /hscrollbar/seeker/button/file**Global EWL_SCROLLPANE_TYPE** (p. 410) /scrollpane/file

/scrollpane/group

Global EWL_SEEKER_TYPE (p. 416) /hseeker/file

/hseeker/group

/vseeker/file

/vseeker/group

Global EWL_SEPARATOR_TYPE (p. 420) /hseparator/file

/hseparator/group

/vseparator/file

/vseparator/group

Global EWL_SHADOW_TYPE (p. 423) /shadow/file

/shadow/group

Global EWL_SPACER_TYPE (p. 425) /spacer/file

/spacer/group

Global EWL_SPINNER_TYPE (p. 433) /spinner/file

/spinner/group

Global EWL_STATUSBAR_TYPE (p. 437) /statusbar/file

/statusbar/group

Global EWL_TABLE_TYPE (p. 446) /table/file

/table/group

Global EWL_TEXT_TYPE (p. 461) /text/file

/text/group

/text/font

/text/font_size

/text/color/r

/text/color/g

/text/color/b

/text/color/a

Global EWL_TREE_TYPE (p. 496) /tree/file

/tree/group

Global EWL_TREE2_TYPE (p. 521) /tree/file

/tree/group

Global EWL_WINDOW_TYPE (p. 566) /window/file

/window/group

Index

- /home/ewww/e17/libs/ewl/TODO, 1291
- UNUSED--
 - ewl_embed_test.c, 796
 - ewl_private.h, 1153
 - ewl_simple_test.c, 797
 - ewl_test_private.h, 803
- a
 - Ewl_Color_Set, 600
- al_align
 - Ewl_Box_Orientation, 588
- a3_align
 - Ewl_Box_Orientation, 588
- A_CALC
 - lib/ewl_histogram.c, 1037
- action_area
 - Ewl_Dialog, 627
- active_area
 - Ewl_Dialog, 627
- ah
 - Ewl_Image, 686
- align
 - Ewl_Text_Context, 761
- align_ask
 - Ewl_Box_Orientation, 588
- align_set
 - Ewl_Box_Orientation, 588
- alpha
 - Ewl_Colorpicker, 604
- alt
 - Ewl_Icon, 684
- app_name
 - Ewl_Config, 614
- appearance
 - Ewl_Widget, 785
- areas
 - Ewl_Text_Trigger, 765
- attach
 - Ewl_Attach_Tooltip, 583
 - Ewl_Widget, 785
- auto_label
 - Ewl_Progressbar, 723
- autohide
 - Ewl_Seeker, 734
- aw
 - Ewl_Image, 686
- b
 - Ewl_Color_Set, 600
 - Ewl_Colorpicker, 604
 - Ewl_Object, 710
- B_CALC
 - lib/ewl_histogram.c, 1037
- backtrace
 - Ewl_Config_Cache, 615
- bar
 - Ewl_Progressbar, 723
- base
 - Ewl_Event_Key_Down, 645
 - Ewl_Event_Key_Up, 646
 - Ewl_Event_Mouse_Down, 648
 - Ewl_Event_Mouse_In, 649
 - Ewl_Event_Mouse_Move, 650
 - Ewl_Event_Mouse_Out, 651
 - Ewl_Event_Mouse_Up, 652
 - Ewl_Event_Mouse_Wheel, 653
 - Ewl_Filelist, 659
- bg
 - Ewl_Text_Context, 761
- bin/ewl_embed_test.c, 795
- bin/ewl_simple_test.c, 797
- bin/Ewl_Test.h, 798
- bin/ewl_test_private.h, 801
- bin/main.c, 804
- bin/tests/border/ewl_border.c, 829
 - test_info, 829
- bin/tests/box/ewl_box.c, 834
 - test_info, 834
- bin/tests/button/ewl_button.c, 839
 - test_info, 839
- bin/tests/calendar/ewl_calendar.c, 844
 - test_info, 844
- bin/tests/colordialog/ewl_colordialog.c, 862
 - test_info, 862
- bin/tests/colorpicker/ewl_colorpicker.c, 869
 - test_info, 869
- bin/tests/combo/ewl_combo.c, 874
 - Combo_Test_Data, 874
 - test_info, 874
- bin/tests/container/ewl_container.c, 885

- test_info, 885
- bin/tests/datepicker/ewl_datepicker.c, 898
 - test_info, 898
- bin/tests/dialog/ewl_dialog.c, 908
 - test_info, 908
- bin/tests/dnd_snoop/ewl_dnd_snoop.c, 806
- bin/tests/engine/ewl_engine.c, 808
- bin/tests/entry/ewl_entry.c, 967
 - test_info, 967
- bin/tests/filedialog/ewl_filedialog.c, 986
 - test_info, 986
- bin/tests/filepicker/ewl_filepicker.c, 1013
 - test_info, 1013
- bin/tests/float/ewl_float.c, 1018
 - test_info, 1018
- bin/tests/freebox/ewl_freebox.c, 1023
 - Freebox_Test, 1023
 - test_info, 1023
- bin/tests/fullscreen/ewl_fullscreen.c, 809
- bin/tests/grid/ewl_grid.c, 1029
 - test_info, 1029
- bin/tests/histogram/ewl_histogram.c, 1038
 - test_info, 1038
- bin/tests/icon/ewl_icon.c, 1044
 - test_info, 1044
- bin/tests/icon_theme/ewl_icon_theme.c, 1049
 - EWL_ICON_THEME_PER_LOOP, 1049
 - icons, 1049
 - test_info, 1049
- bin/tests/image/ewl_image.c, 1060
 - test_info, 1060
- bin/tests/image_thumbnail/ewl_image_thumbnail.c, 810
- bin/tests/io_manager/ewl_io_manager.c, 1068
 - test_info, 1068
- bin/tests/layer/ewl_layer.c, 811
- bin/tests/list/ewl_list.c, 1077
 - List_Test_Data, 1077
 - List_Test_Row_Data, 1077
 - test_info, 1077
- bin/tests/media/ewl_media.c, 1084
 - PATH_MAX, 1084
 - test_info, 1084
- bin/tests/menu/ewl_menu.c, 1089
 - test_info, 1089
- bin/tests/menubar/ewl_menubar.c, 1096
 - test_info, 1096
- bin/tests/modal/ewl_modal.c, 812
- bin/tests/mvc/ewl_mvc.c, 1112
 - test_info, 1112
- bin/tests/notebook/ewl_notebook.c, 1119
 - test_info, 1119
- bin/tests/paned/ewl_paned.c, 1137
 - test_info, 1137
- bin/tests/password/ewl_password.c, 1142
 - test_info, 1142
- bin/tests/pointer/ewl_pointer.c, 813
- bin/tests/progressbar/ewl_progressbar.c, 1156
 - test_info, 1156
- bin/tests/puzzle/ewl_puzzle.c, 814
- bin/tests/scrollbar/ewl_scrollbar.c, 1172
 - test_info, 1172
- bin/tests/scrollpane/ewl_scrollpane.c, 1177
 - test_info, 1177
- bin/tests/seeker/ewl_seeker.c, 1182
 - test_info, 1182
- bin/tests/shadow/ewl_shadow.c, 1189
 - test_info, 1189
- bin/tests/spinner/ewl_spinner.c, 1199
 - test_info, 1199
- bin/tests/statusbar/ewl_statusbar.c, 1204
 - test_info, 1204
- bin/tests/table/ewl_table.c, 1213
 - test_info, 1213
- bin/tests/text/ewl_text.c, 1225
 - test_info, 1225
- bin/tests/text_editor/ewl_text_editor.c, 815
- bin/tests/theme/ewl_theme.c, 1236
 - test_info, 1236
- bin/tests/toolbar/ewl_toolbar.c, 1239
 - cb_click, 1239
 - test_info, 1239
- bin/tests/tooltip/ewl_tooltip.c, 816
- bin/tests/tree/ewl_tree.c, 1246
 - COLS, 1246
 - NEST, 1246
 - ROWS, 1246
 - test_info, 1246
- bin/tests/tree2/ewl_tree2.c, 1253
 - test_info, 1254
 - TREE2_DATA_ELEMENTS, 1254
 - Tree2_Test_Data, 1254
 - Tree2_Test_Row_Data, 1254
- bin/tests/widget/ewl_widget.c, 1277
 - test_info, 1277
- block_seek
 - Ewl_Media, 694
- body
 - Ewl_Border, 584
 - Ewl_Button, 589
 - Ewl_Notebook, 707
- box
 - Ewl_Attach_Tooltip, 583
 - Ewl_Border, 584
 - Ewl_Calendar, 591

- Ewl_Colorpicker, 604
- Ewl_Dialog, 627
- Ewl_Filelist, 659
- Ewl_Filepicker, 668
- Ewl_Floater, 671
- Ewl_MVC, 705
- Ewl_Notebook, 707
- Ewl_Scrollbar, 730
- Ewl_Scrollpane, 732
- Ewl_Shadow, 741
- Ewl_Stock, 750
- Ewl_Tree2_View, 776
- Ewl_Tree2_View_Scrolled, 779
- buffer
 - Ewl_Configure_Queue, 617
- built_children
 - Ewl_Tree2_Node, 774
- button
 - Ewl_Checkbutton, 598
 - Ewl_Colorpicker_Radiobutton, 607
 - Ewl_Combo, 610
 - Ewl_Event_Mouse_Down, 648
 - Ewl_Event_Mouse_Up, 652
 - Ewl_Menu_Item, 699
 - Ewl_Radiobutton, 725
 - Ewl_Seeker, 734
- buttons_alignment
 - Ewl_Scrollbar, 730
- byte_idx
 - Ewl_Text, 758
- byte_len
 - Ewl_Text_Fmt, 764
- bytes
 - Ewl_Text, 758
- calendar
 - Ewl_Datepicker, 624
- calendar_window
 - Ewl_Datepicker, 624
- callbacks
 - Ewl_Widget, 785
- canvas
 - Ewl_Embed, 630
 - Ewl_Engine_Info, 634
 - Ewl_Spectrum, 744
- canvas_window
 - Ewl_Embed, 630
- cb
 - Control, 578
 - Ewl_Callback_Custom, 595
 - Ewl_Menu_Info, 698
 - Ewl_MVC, 705
- cb_click
 - bin/tests/toolbar/ewl_toolbar.c, 1239
- cell
 - Ewl_Combo_Cell, 612
- chain
 - Ewl_Radiobutton, 725
- channel
 - Ewl_Histogram, 681
- char_base
 - Ewl_Text_Trigger, 765
- char_idx
 - Ewl_Text, 758
- char_len
 - Ewl_Text_Fmt, 764
 - Ewl_Text_Trigger, 765
- char_pos
 - Ewl_Text_Trigger, 766
- chars
 - Ewl_Text, 758
- check
 - Ewl_Checkbutton, 598
- checked
 - Ewl_Check, 597
- child_add
 - Ewl_Container, 618
 - Ewl_Context_Menu, 621
- child_hide
 - Ewl_Container, 618
- child_remove
 - Ewl_Container, 618
 - Ewl_Context_Menu, 621
- child_resize
 - Ewl_Container, 618
- child_show
 - Ewl_Container, 619
- children
 - Ewl_Container, 619
- classname
 - Ewl_Window, 789
- clicked
 - Ewl_Embed, 630
- clicks
 - Ewl_Event_Mouse_Down, 648
- clip_box
 - Ewl_Container, 619
- col_headers
 - Ewl_Table, 753
- col_size
 - Ewl_Grid, 675
- color
 - Ewl_Histogram, 681
 - Ewl_Text_Context, 762
- COLOR_HASH
 - lib/ewl_text.c, 1223
- COLS
 - bin/tests/tree/ewl_tree.c, 1246

- cols
 - Ewl_Grid, 675
- column
 - Ewl_Selection_Idx, 737
 - Ewl_Selection_Range, 738
 - Ewl_Tree2, 772
- columns
 - Ewl_Tree2, 772
- combo
 - Ewl_Combo_Cell, 612
- Combo_Test_Data, 577
 - bin/tests/combo/ewl_combo.c, 874
 - count, 577
 - data, 577
- comparator
 - Ewl_Freebox, 673
- compare
 - Freebox_Test, 790
- complex_label
 - Ewl_Icon, 684
- compress_label
 - Ewl_Icon, 684
- configure_active
 - ewl_misc.c, 1101
- configure_available
 - ewl_misc.c, 1101
- CONFIGURED
 - Ewl_Object, 347
- container
 - Ewl_Box, 586
 - Ewl_Cell, 596
 - Ewl_Context_Menu, 621
 - Ewl_Freebox, 673
 - Ewl_Grid, 675
 - Ewl_Overlay, 712
 - Ewl_Paned, 715
 - Ewl_Range, 727
 - Ewl_Row, 729
 - Ewl_Scrollpane, 732
 - Ewl_Table, 753
 - Ewl_Text, 758
 - Ewl_Tree, 769
 - Ewl_Tree_Node, 781
- content_view
 - Ewl_Tree2, 772
- Control, 578
 - cb, 578
 - func, 578
 - name, 578
- count
 - Combo_Test_Data, 577
 - Ewl_Model, 703
 - List_Test_Data, 791
 - Tree2_Test_Data, 793
- cross_hairs
 - Ewl_Spectrum, 744
- cs
 - Ewl_Image, 687
- cur_day
 - Ewl_Calendar, 591
- cur_month
 - Ewl_Calendar, 591
- cur_page
 - Ewl_Notebook, 707
- cur_year
 - Ewl_Calendar, 592
- current
 - Ewl_Colorpicker, 604
 - Ewl_Object, 710
 - Ewl_Statusbar, 748
 - Ewl_Text, 758
- CURRENT_H
 - Ewl_Object, 347
- current_pos
 - Ewl_Grid_Info, 678
- current_position
 - Ewl_Paned_Layout, 718
- current_size
 - Ewl_Grid_Info, 678
 - Ewl_Paned_Layout, 718
- CURRENT_W
 - Ewl_Object, 348
- CURRENT_X
 - Ewl_Object, 348
- CURRENT_Y
 - Ewl_Object, 348
- cursor
 - Ewl_Attach_Dnd, 580
 - Ewl_Embed, 630
 - Ewl_Entry, 635
- cursor_position
 - Ewl_Text, 758
- data
 - Combo_Test_Data, 577
 - Ewl_Attach, 579
 - Ewl_Attach_Dnd, 580
 - Ewl_Config, 614
 - Ewl_Event_Dnd_Data_Received, 640
 - Ewl_Event_Dnd_Drop, 642
 - Ewl_Model, 703
 - Ewl_MVC, 706
 - Ewl_Selection, 736
- data_dirty
 - Ewl_Grid, 676
- data_free
 - Ewl_Model, 703
- data_type

- Ewl_Attach, 579
- DCHECK_PARAM_PTR
 - ewl_debug.h, 903
- DCHECK_PARAM_PTR_RET
 - ewl_debug.h, 903
- DCHECK_TYPE
 - ewl_debug.h, 903
- DCHECK_TYPE_RET
 - ewl_debug.h, 903
- decrement
 - Ewl_Scrollbar, 730
 - Ewl_Spinner, 746
- delete_count
 - Ewl_Text, 758
- deleted
 - Ewl_Text_Trigger_Area, 767
- DENTER_FUNCTION
 - ewl_debug.h, 903
- dependancies
 - Ewl_Engine, 633
- DERROR
 - ewl_debug.h, 903
- DESTROYED
 - Ewl_Object, 348
- dialog
 - Ewl_Colordialog, 601
 - Ewl_Filedialog, 657
- digits
 - Ewl_Spinner, 746
- dir
 - Ewl_Event_Mouse_Wheel, 653
 - Ewl_Filelist_Column_Data, 663
- dir_change
 - Ewl_Filelist, 659
- direct
 - Ewl_Attach_List, 581
 - Ewl_Pair_List, 714
- direction
 - Ewl_Scrollbar, 731
 - Ewl_Spinner, 747
 - Ewl_Tree2, 772
- directory
 - Ewl_Filelist, 659
- dirs
 - Ewl_Filelist_Column, 661
- dirty
 - Ewl_MVC, 706
 - Ewl_Spectrum, 744
 - Ewl_Text, 758
- DISABLED
 - Ewl_Object, 348
- DLEAVE_FUNCTION
 - ewl_debug.h, 903
- DLEVEL_STABLE
 - ewl_debug.h, 904
- DLEVEL_TESTING
 - ewl_debug.h, 904
- DLEVEL_UNSTABLE
 - ewl_debug.h, 904
- dnd_count
 - Ewl_Embed, 630
- dnd_last_position
 - Ewl_Embed, 630
- dnd_types
 - Ewl_Embed, 630
- double_underline
 - Ewl_Text_Context, 762
- drag_widget
 - Ewl_Embed, 630
- dragstart
 - Ewl_Seeker, 734
- DRETURN
 - ewl_debug.h, 904
- DRETURN_FLOAT
 - ewl_debug.h, 904
- DRETURN_INT
 - ewl_debug.h, 904
- DRETURN_PTR
 - ewl_debug.h, 904
- drop_widget
 - Ewl_Embed, 630
- DWARNING
 - ewl_debug.h, 904
- editable
 - Ewl_Combo, 610
 - Ewl_Entry, 635
 - Ewl_Icon, 684
- edje
 - ewl_embed_test.c, 796
- ee
 - ewl_embed_test.c, 796
- embed
 - Ewl_Attach_Tooltip, 583
 - Ewl_Window, 789
- enable
 - Ewl_Config_Cache, 615
- end
 - Ewl_Configure_Queue, 617
 - Ewl_Selection_Range, 738
- end_c
 - Ewl_Table, 754
- end_col
 - Ewl_Grid_Child, 677
- end_r
 - Ewl_Table, 754
- end_row
 - Ewl_Grid_Child, 677

- engine
 - Ewl_Embed, 630
- engine_name
 - Ewl_Embed, 631
- entry
 - ewl_embed_test.c, 796
 - Ewl_Password, 719
 - Ewl_Spinner, 747
- ev_clip
 - Ewl_Embed, 631
- evas
 - ewl_embed_test.c, 796
- evas_render
 - Ewl_Config_Cache, 615
- event_id
 - Ewl_Callback_Custom, 595
- ewl
 - Ewl_Window, 789
- EWL_ATTACH
 - Ewl_Attach, 22
- Ewl_Attach, 579
 - data, 579
 - data_type, 579
 - EWL_ATTACH, 22
 - Ewl_Attach, 24
 - ewl_attach_color_get, 22
 - ewl_attach_color_set, 22
 - Ewl_Attach_Dnd, 24
 - ewl_attach_get, 24
 - EWL_ATTACH_IS, 22
 - ewl_attach_list_del, 24
 - ewl_attach_mouse_argb_cursor_set, 23
 - ewl_attach_mouse_cursor_set, 23
 - ewl_attach_name_get, 23
 - ewl_attach_name_set, 23
 - ewl_attach_other_set, 24
 - ewl_attach_text_set, 25
 - ewl_attach_tooltip_text_set, 23
 - EWL_ATTACH_TOOLTIP_TYPE, 23
 - ewl_attach_tooltip_widget_set, 23
 - ewl_attach_widget_association_get, 23
 - ewl_attach_widget_association_set, 24
 - ewl_attach_widget_set, 25
 - priv_data, 579
 - type, 579
- ewl_attach.c
 - ewl_attach_dnd_drag_set, 822
 - Ewl_Attach_Tooltip, 822
- Ewl_Attach: A mechanism to attach data to
 - widgets., 21
- ewl_attach_color_get
 - Ewl_Attach, 22
- ewl_attach_color_set
 - Ewl_Attach, 22
- Ewl_Attach_Data_Type
 - Ewl_Enums, 151, 154
- EWL_ATTACH_DATA_TYPE_OTHER
 - Ewl_Enums, 154
- EWL_ATTACH_DATA_TYPE_TEXT
 - Ewl_Enums, 154
- EWL_ATTACH_DATA_TYPE_WIDGET
 - Ewl_Enums, 154
- Ewl_Attach_Dnd, 580
 - cursor, 580
 - data, 580
 - Ewl_Attach, 24
 - size, 580
- ewl_attach_dnd_drag_set
 - ewl_attach.c, 822
- ewl_attach_get
 - Ewl_Attach, 24
- EWL_ATTACH_IS
 - Ewl_Attach, 22
- Ewl_Attach_List, 581
 - direct, 581
 - Ewl_Widget, 546
 - len, 581
 - list, 581
- ewl_attach_list_del
 - Ewl_Attach, 24
- ewl_attach_mouse_argb_cursor_set
 - Ewl_Attach, 23
- ewl_attach_mouse_cursor_set
 - Ewl_Attach, 23
- ewl_attach_name_get
 - Ewl_Attach, 23
- ewl_attach_name_set
 - Ewl_Attach, 23
- ewl_attach_other_set
 - Ewl_Attach, 24
- ewl_attach_text_set
 - Ewl_Attach, 25
- Ewl_Attach_Tooltip, 582
 - attach, 583
 - box, 583
 - embed, 583
 - ewl_attach.c, 822
 - timer, 583
 - to, 583
 - win, 583
 - x, 583
 - y, 583
- ewl_attach_tooltip_text_set
 - Ewl_Attach, 23
- EWL_ATTACH_TOOLTIP_TYPE
 - Ewl_Attach, 23
- ewl_attach_tooltip_widget_set
 - Ewl_Attach, 23

- Ewl_Attach_Type
 - Ewl_Enums, 151, 154
- EWL_ATTACH_TYPE_COLOR
 - Ewl_Enums, 154
- EWL_ATTACH_TYPE_DND_DATA
 - Ewl_Enums, 154
- EWL_ATTACH_TYPE_MOUSE_ARGB_ -
CURSOR
 - Ewl_Enums, 154
- EWL_ATTACH_TYPE_MOUSE_CURSOR
 - Ewl_Enums, 154
- EWL_ATTACH_TYPE_NAME
 - Ewl_Enums, 154
- EWL_ATTACH_TYPE_TOOLTIP
 - Ewl_Enums, 154
- EWL_ATTACH_TYPE_WIDGET_ -
ASSOCIATION
 - Ewl_Enums, 154
- ewl_attach_widget_association_get
 - Ewl_Attach, 23
- ewl_attach_widget_association_set
 - Ewl_Attach, 24
- ewl_attach_widget_set
 - Ewl_Attach, 25
- ewl_backtrace
 - ewl_debug.h, 904
 - ewl_misc.c, 1101
- EWL_BORDER
 - Ewl_Border, 27
- Ewl_Border, 584
 - body, 584
 - box, 584
 - EWL_BORDER, 27
 - Ewl_Border, 27
 - ewl_border_init, 27
 - EWL_BORDER_IS, 27
 - ewl_border_label_alignment_get, 27
 - ewl_border_label_alignment_set, 27
 - ewl_border_label_get, 28
 - ewl_border_label_position_get, 28
 - ewl_border_label_position_set, 28
 - ewl_border_label_set, 28
 - ewl_border_new, 29
 - EWL_BORDER_TYPE, 27
 - label, 584
 - label_position, 584
- Ewl_Border: A container with a border and la-
bel, 26
- ewl_border_init
 - Ewl_Border, 27
- EWL_BORDER_IS
 - Ewl_Border, 27
- ewl_border_label_alignment_get
 - Ewl_Border, 27
- ewl_border_label_alignment_set
 - Ewl_Border, 27
- ewl_border_label_get
 - Ewl_Border, 28
- ewl_border_label_position_get
 - Ewl_Border, 28
- ewl_border_label_position_set
 - Ewl_Border, 28
- ewl_border_label_set
 - Ewl_Border, 28
- ewl_border_new
 - Ewl_Border, 29
- EWL_BORDER_TYPE
 - Ewl_Border, 27
- EWL_BOX
 - Ewl_Box, 31
- Ewl_Box, 586
 - container, 586
 - EWL_BOX, 31
 - Ewl_Box, 32
 - ewl_box_cb_child_hide, 32
 - ewl_box_cb_child_homogeneous_show,
32
 - ewl_box_cb_child_resize, 32
 - ewl_box_cb_child_show, 32
 - ewl_box_cb_configure, 32
 - ewl_box_cb_configure_homogeneous, 32
 - ewl_box_homogeneous_set, 32
 - ewl_box_init, 32
 - EWL_BOX_IS, 31
 - ewl_box_new, 32
 - ewl_box_orientation_get, 33
 - ewl_box_orientation_set, 33
 - ewl_box_spacing_set, 33
 - EWL_BOX_TYPE, 31
 - ewl_hbox_new, 33
 - ewl_vbox_new, 34
 - homogeneous, 586
 - orientation, 586
 - spacing, 586
- Ewl_Box: The Box Layout Container., 30
- ewl_box_cb_child_hide
 - Ewl_Box, 32
- ewl_box_cb_child_homogeneous_show
 - Ewl_Box, 32
 - lib/ewl_box.c, 833
- ewl_box_cb_child_resize
 - Ewl_Box, 32
 - lib/ewl_box.c, 833
- ewl_box_cb_child_show
 - Ewl_Box, 32
- ewl_box_cb_configure
 - Ewl_Box, 32
 - lib/ewl_box.c, 833

- ewl_box_cb_configure_homogeneous
 - Ewl_Box, 32
 - lib/ewl_box.c, 833
- ewl_box_homogeneous_set
 - Ewl_Box, 32
- ewl_box_init
 - Ewl_Box, 32
- EWL_BOX_IS
 - Ewl_Box, 31
- ewl_box_new
 - Ewl_Box, 32
- Ewl_Box_Orientation, 588
 - a1_align, 588
 - a3_align, 588
 - align_ask, 588
 - align_set, 588
 - f1_align, 588
 - f3_align, 588
 - f_policy, 588
 - fill_ask, 588
 - fill_set, 588
 - pref_fill_set, 588
- ewl_box_orientation_get
 - Ewl_Box, 33
- ewl_box_orientation_set
 - Ewl_Box, 33
- ewl_box_spacing_set
 - Ewl_Box, 33
- EWL_BOX_TYPE
 - Ewl_Box, 31
- EWL_BUTTON
 - Ewl_Button, 36
- Ewl_Button, 589
 - body, 589
 - EWL_BUTTON, 36
 - Ewl_Button, 37
 - ewl_button_alignment_get, 37
 - ewl_button_alignment_set, 37
 - ewl_button_cb_key_down, 37
 - ewl_button_fill_policy_get, 37
 - ewl_button_fill_policy_set, 37
 - ewl_button_image_get, 38
 - ewl_button_image_set, 38
 - ewl_button_image_size_get, 38
 - ewl_button_image_size_set, 38
 - ewl_button_init, 39
 - EWL_BUTTON_IS, 36
 - ewl_button_label_get, 39
 - ewl_button_label_set, 39
 - ewl_button_new, 39
 - EWL_BUTTON_TYPE, 36
 - ewl_button_view_get, 40
 - image_object, 589
 - label_object, 589
 - stock, 590
- Ewl_Button: The Basic Button, 35
- ewl_button_alignment_get
 - Ewl_Button, 37
- ewl_button_alignment_set
 - Ewl_Button, 37
- ewl_button_cb_key_down
 - Ewl_Button, 37
 - lib/ewl_button.c, 838
- ewl_button_fill_policy_get
 - Ewl_Button, 37
- ewl_button_fill_policy_set
 - Ewl_Button, 37
- ewl_button_image_get
 - Ewl_Button, 38
- ewl_button_image_set
 - Ewl_Button, 38
- ewl_button_image_size_get
 - Ewl_Button, 38
- ewl_button_image_size_set
 - Ewl_Button, 38
- ewl_button_init
 - Ewl_Button, 39
- EWL_BUTTON_IS
 - Ewl_Button, 36
- ewl_button_label_get
 - Ewl_Button, 39
- ewl_button_label_set
 - Ewl_Button, 39
- ewl_button_new
 - Ewl_Button, 39
- EWL_BUTTON_TYPE
 - Ewl_Button, 36
- ewl_button_view_get
 - Ewl_Button, 40
- EWL_CALENDAR
 - Ewl_Calendar, 42
- Ewl_Calendar, 591
 - box, 591
 - cur_day, 591
 - cur_month, 591
 - cur_year, 592
 - EWL_CALENDAR, 42
 - Ewl_Calendar, 42
 - ewl_calendar_ascii_time_get, 42
 - ewl_calendar_day_get, 42
 - ewl_calendar_init, 42
 - EWL_CALENDAR_IS, 42
 - ewl_calendar_month_get, 43
 - ewl_calendar_new, 43
 - EWL_CALENDAR_TYPE, 42
 - ewl_calendar_year_get, 43
 - grid, 592
 - month_label, 592

- Ewl_Calendar: The EWL Calendar Widget, 41
- ewl_calendar_ascii_time_get
 - Ewl_Calendar, 42
- ewl_calendar_day_get
 - Ewl_Calendar, 42
- ewl_calendar_init
 - Ewl_Calendar, 42
- EWL_CALENDAR_IS
 - Ewl_Calendar, 42
- ewl_calendar_month_get
 - Ewl_Calendar, 43
- ewl_calendar_new
 - Ewl_Calendar, 43
- EWL_CALENDAR_TYPE
 - Ewl_Calendar, 42
- ewl_calendar_year_get
 - Ewl_Calendar, 43
- EWL_CALLBACK
 - Ewl_Callback, 45
- Ewl_Callback, 593
 - EWL_CALLBACK, 45
 - Ewl_Callback, 47
 - ewl_callback_append, 47
 - ewl_callback_call, 48
 - ewl_callback_call_with_event_data, 48
 - ewl_callback_clear, 48
 - EWL_CALLBACK_CUSTOM, 45
 - Ewl_Callback_Custom, 47
 - ewl_callback_del, 48
 - ewl_callback_del_type, 48
 - ewl_callback_del_with_data, 49
 - EWL_CALLBACK_FLAG_-
INTERCEPT, 45
 - EWL_CALLBACK_FLAG_-
NOINTERCEPT, 46
 - EWL_CALLBACK_FLAG_NOTIFY, 46
 - EWL_CALLBACK_FLAGS, 46
 - EWL_CALLBACK_FUNCTION, 46
 - Ewl_Callback_Function, 47
 - EWL_CALLBACK_INDEX, 46
 - ewl_callback_insert_after, 49
 - EWL_CALLBACK_LEN, 46
 - EWL_CALLBACK_LIST, 46
 - EWL_CALLBACK_NOTIFY_MASK,
46
 - EWL_CALLBACK_POS, 46
 - ewl_callback_prepend, 49
 - EWL_CALLBACK_SET_DIRECT, 47
 - EWL_CALLBACK_SET_NODIRECT,
47
 - ewl_callback_type_add, 50
 - ewl_callbacks_init, 50
 - ewl_callbacks_shutdown, 50
 - func, 593
 - id, 593
 - references, 593
 - user_data, 593
- ewl_callback.c
 - ewl_callback_del_cb_id, 848
- Ewl_Callback: The Callback Mechanisms, 44
- ewl_callback_append
 - Ewl_Callback, 47
- ewl_callback_call
 - Ewl_Callback, 48
- ewl_callback_call_with_event_data
 - Ewl_Callback, 48
- Ewl_Callback_Chain, 594
 - Ewl_Widget, 546
 - index, 594
 - len, 594
 - list, 594
 - mask, 594
- ewl_callback_clear
 - Ewl_Callback, 48
- EWL_CALLBACK_CLICKED
 - Ewl_Enums, 155
- EWL_CALLBACK_CONFIGURE
 - Ewl_Enums, 154
- EWL_CALLBACK_CUSTOM
 - Ewl_Callback, 45
- Ewl_Callback_Custom, 595
 - cb, 595
 - event_id, 595
 - Ewl_Callback, 47
- ewl_callback_del
 - Ewl_Callback, 48
- ewl_callback_del_cb_id
 - ewl_callback.c, 848
- ewl_callback_del_type
 - Ewl_Callback, 48
- ewl_callback_del_with_data
 - Ewl_Callback, 49
- EWL_CALLBACK_DELETE_WINDOW
 - Ewl_Window, 575
- EWL_CALLBACK_DESTROY
 - Ewl_Enums, 154
- EWL_CALLBACK_DND_DATA_-
RECEIVED
 - Ewl_Dnd, 117
- EWL_CALLBACK_DND_DATA_-
REQUEST
 - Ewl_Dnd, 117
- EWL_CALLBACK_DND_DROP
 - Ewl_Dnd, 117
- EWL_CALLBACK_DND_ENTER
 - Ewl_Dnd, 117
- EWL_CALLBACK_DND_LEAVE
 - Ewl_Dnd, 117

- EWL_CALLBACK_DND_POSITION
 - Ewl_Dnd, 117
- EWL_CALLBACK_EXPOSE
 - Ewl_Window, 575
- EWL_CALLBACK_FLAG_INTERCEPT
 - Ewl_Callback, 45
- EWL_CALLBACK_FLAG_NOINTERCEPT
 - Ewl_Callback, 46
- EWL_CALLBACK_FLAG_NOTIFY
 - Ewl_Callback, 46
- EWL_CALLBACK_FLAGS
 - Ewl_Callback, 46
- EWL_CALLBACK_FOCUS_IN
 - Ewl_Enums, 155
- EWL_CALLBACK_FOCUS_OUT
 - Ewl_Enums, 155
- EWL_CALLBACK_FUNCTION
 - Ewl_Callback, 46
- Ewl_Callback_Function
 - Ewl_Callback, 47
- EWL_CALLBACK_HIDE
 - Ewl_Enums, 154
- EWL_CALLBACK_INDEX
 - Ewl_Callback, 46
- ewl_callback_insert_after
 - Ewl_Callback, 49
- EWL_CALLBACK_KEY_DOWN
 - Ewl_Enums, 155
- EWL_CALLBACK_KEY_UP
 - Ewl_Enums, 155
- EWL_CALLBACK_LEN
 - Ewl_Callback, 46
- EWL_CALLBACK_LIST
 - Ewl_Callback, 46
- EWL_CALLBACK_MAX
 - Ewl_Enums, 155
- EWL_CALLBACK_MOUSE_DOWN
 - Ewl_Enums, 155
- EWL_CALLBACK_MOUSE_IN
 - Ewl_Enums, 155
- EWL_CALLBACK_MOUSE_MOVE
 - Ewl_Enums, 155
- EWL_CALLBACK_MOUSE_OUT
 - Ewl_Enums, 155
- EWL_CALLBACK_MOUSE_UP
 - Ewl_Enums, 155
- EWL_CALLBACK_MOUSE_WHEEL
 - Ewl_Enums, 155
- EWL_CALLBACK_NOTIFY_INTERCEPT
 - Ewl_Enums, 155
- EWL_CALLBACK_NOTIFY_MASK
 - Ewl_Callback, 46
- EWL_CALLBACK_NOTIFY_NONE
 - Ewl_Enums, 155
- EWL_CALLBACK_NOTIFY_NOTIFY
 - Ewl_Enums, 155
- EWL_CALLBACK_OBSCURE
 - Ewl_Enums, 154
- EWL_CALLBACK_POS
 - Ewl_Callback, 46
- ewl_callback_prepend
 - Ewl_Callback, 49
- EWL_CALLBACK_REALIZE
 - Ewl_Enums, 154
- EWL_CALLBACK_REPARENT
 - Ewl_Enums, 155
- EWL_CALLBACK_REVEAL
 - Ewl_Enums, 154
- EWL_CALLBACK_SET_DIRECT
 - Ewl_Callback, 47
- EWL_CALLBACK_SET_NODIRECT
 - Ewl_Callback, 47
- EWL_CALLBACK_SHOW
 - Ewl_Enums, 154
- EWL_CALLBACK_STATE_CHANGED
 - Ewl_Enums, 155
- Ewl_Callback_Type
 - Ewl_Enums, 151, 154
- ewl_callback_type_add
 - Ewl_Callback, 50
- EWL_CALLBACK_TYPE_DIRECT
 - Ewl_Enums, 155
- EWL_CALLBACK_UNREALIZE
 - Ewl_Enums, 154
- EWL_CALLBACK_VALUE_CHANGED
 - Ewl_Enums, 155
- EWL_CALLBACK_WIDGET_DISABLE
 - Ewl_Enums, 155
- EWL_CALLBACK_WIDGET_ENABLE
 - Ewl_Enums, 155
- ewl_callbacks_init
 - Ewl_Callback, 50
- ewl_callbacks_shutdown
 - Ewl_Callback, 50
- ewl_canvas_destroy
 - Ewl_Misc, 313
- ewl_canvas_object_destroy
 - Ewl_Misc, 313
- EWL_CELL
 - Ewl_Cell, 51
- Ewl_Cell, 596
 - container, 596
- EWL_CELL, 51
 - Ewl_Cell, 52
- ewl_cell_cb_child_resize, 52
- ewl_cell_cb_child_show, 52
- ewl_cell_cb_configure, 52
- ewl_cell_init, 52

- EWL_CELL_IS, 51
- ewl_cell_new, 52
- EWL_CELL_TYPE, 52
- ewl_cell.c
 - ewl_cell_cb_configure, 851
- Ewl_Cell: The Cell Container, Layout for a Single Widget, 51
- ewl_cell_cb_child_resize
 - Ewl_Cell, 52
- ewl_cell_cb_child_show
 - Ewl_Cell, 52
- ewl_cell_cb_configure
 - Ewl_Cell, 52
 - ewl_cell.c, 851
- ewl_cell_init
 - Ewl_Cell, 52
- EWL_CELL_IS
 - Ewl_Cell, 51
- ewl_cell_new
 - Ewl_Cell, 52
- EWL_CELL_TYPE
 - Ewl_Cell, 52
- EWL_CHECK
 - Ewl_Check, 53
- Ewl_Check, 597
 - checked, 597
 - EWL_CHECK, 53
 - Ewl_Check, 54
 - ewl_check_cb_clicked, 54
 - ewl_check_cb_update_check, 54
 - ewl_check_checked_set, 54
 - ewl_check_init, 54
 - EWL_CHECK_IS, 53
 - ewl_check_is_checked, 54
 - ewl_check_new, 55
 - EWL_CHECK_TYPE, 54
 - w, 597
- ewl_check.c
 - ewl_check_cb_clicked, 853
 - ewl_check_cb_update_check, 853
- Ewl_Check: The Simple Check for a Checkbutton, 53
- ewl_check_cb_clicked
 - Ewl_Check, 54
 - ewl_check.c, 853
- ewl_check_cb_update_check
 - Ewl_Check, 54
 - ewl_check.c, 853
- ewl_check_checked_set
 - Ewl_Check, 54
- ewl_check_init
 - Ewl_Check, 54
- EWL_CHECK_IS
 - Ewl_Check, 53
- ewl_check_is_checked
 - Ewl_Check, 54
- ewl_check_new
 - Ewl_Check, 55
- EWL_CHECK_TYPE
 - Ewl_Check, 54
- EWL_CHECKBUTTON
 - Ewl_Checkbutton, 57
- Ewl_Checkbutton, 598
 - button, 598
 - check, 598
 - EWL_CHECKBUTTON, 57
 - Ewl_Checkbutton, 57
 - ewl_checkbutton_cb_clicked, 57
 - ewl_checkbutton_checked_set, 57
 - ewl_checkbutton_init, 57
 - EWL_CHECKBUTTON_IS, 57
 - ewl_checkbutton_is_checked, 57
 - ewl_checkbutton_label_position_set, 58
 - ewl_checkbutton_new, 58
 - EWL_CHECKBUTTON_TYPE, 57
 - label_position, 598
- ewl_checkbutton.c
 - ewl_checkbutton_cb_clicked, 857
- Ewl_Checkbutton: An Ewl_Checkbutton with Label, 56
- ewl_checkbutton_cb_clicked
 - Ewl_Checkbutton, 57
 - ewl_checkbutton.c, 857
- ewl_checkbutton_checked_set
 - Ewl_Checkbutton, 57
- ewl_checkbutton_init
 - Ewl_Checkbutton, 57
- EWL_CHECKBUTTON_IS
 - Ewl_Checkbutton, 57
- ewl_checkbutton_is_checked
 - Ewl_Checkbutton, 57
- ewl_checkbutton_label_position_set
 - Ewl_Checkbutton, 58
- ewl_checkbutton_new
 - Ewl_Checkbutton, 58
- EWL_CHECKBUTTON_TYPE
 - Ewl_Checkbutton, 57
- Ewl_Child_Add
 - Ewl_Container, 88
- Ewl_Child_Hide
 - Ewl_Container, 88
- Ewl_Child_Remove
 - Ewl_Container, 88
- Ewl_Child_Resize
 - Ewl_Container, 88
- Ewl_Child_Show
 - Ewl_Container, 88
- Ewl_Color_Mode

- Ewl_Enums, 151, 155
- EWL_COLOR_MODE_HSV_HUE
 - Ewl_Enums, 155
- EWL_COLOR_MODE_HSV_ - SATURATION
 - Ewl_Enums, 155
- EWL_COLOR_MODE_HSV_VALUE
 - Ewl_Enums, 155
- EWL_COLOR_MODE_RGB_BLUE
 - Ewl_Enums, 155
- EWL_COLOR_MODE_RGB_GREEN
 - Ewl_Enums, 155
- EWL_COLOR_MODE_RGB_RED
 - Ewl_Enums, 155
- Ewl_Color_Set, 600
 - a, 600
 - b, 600
 - Ewl_Widget, 546
 - g, 600
 - r, 600
- EWL_COLORDIALOG
 - Ewl_Colordialog, 60
- Ewl_Colordialog, 601
 - dialog, 601
 - EWL_COLORDIALOG, 60
 - Ewl_Colordialog, 60
 - ewl_colordialog_alpha_get, 60
 - ewl_colordialog_alpha_set, 61
 - ewl_colordialog_cb_button_click, 61
 - ewl_colordialog_cb_delete_window, 61
 - ewl_colordialog_color_mode_get, 61
 - ewl_colordialog_color_mode_set, 61
 - ewl_colordialog_current_rgb_get, 61
 - ewl_colordialog_current_rgb_set, 62
 - ewl_colordialog_has_alpha_get, 62
 - ewl_colordialog_has_alpha_set, 62
 - ewl_colordialog_init, 63
 - EWL_COLORDIALOG_IS, 60
 - ewl_colordialog_new, 63
 - ewl_colordialog_previous_rgb_get, 63
 - ewl_colordialog_previous_rgb_set, 63
 - EWL_COLORDIALOG_TYPE, 60
 - picker, 601
- Ewl_Colordialog: A Simple Colour Dialog wid-
get, 59
- ewl_colordialog_alpha_get
 - Ewl_Colordialog, 60
- ewl_colordialog_alpha_set
 - Ewl_Colordialog, 61
- ewl_colordialog_cb_button_click
 - Ewl_Colordialog, 61
 - lib/ewl_colordialog.c, 861
- ewl_colordialog_cb_delete_window
 - Ewl_Colordialog, 61
 - lib/ewl_colordialog.c, 861
- ewl_colordialog_color_mode_get
 - Ewl_Colordialog, 61
- ewl_colordialog_color_mode_set
 - Ewl_Colordialog, 61
- ewl_colordialog_current_rgb_get
 - Ewl_Colordialog, 61
- ewl_colordialog_current_rgb_set
 - Ewl_Colordialog, 62
- ewl_colordialog_has_alpha_get
 - Ewl_Colordialog, 62
- ewl_colordialog_has_alpha_set
 - Ewl_Colordialog, 62
- ewl_colordialog_init
 - Ewl_Colordialog, 63
- EWL_COLORDIALOG_IS
 - Ewl_Colordialog, 60
- ewl_colordialog_new
 - Ewl_Colordialog, 63
- ewl_colordialog_previous_rgb_get
 - Ewl_Colordialog, 63
- ewl_colordialog_previous_rgb_set
 - Ewl_Colordialog, 63
- EWL_COLORDIALOG_TYPE
 - Ewl_Colordialog, 60
- EWL_COLORPICKER
 - Ewl_Colorpicker, 67
- Ewl_Colorpicker, 603
 - alpha, 604
 - b, 604
 - box, 604
 - current, 604
 - EWL_COLORPICKER, 67
 - Ewl_Colorpicker, 67
 - ewl_colorpicker_alpha_get, 67
 - ewl_colorpicker_alpha_set, 67
 - ewl_colorpicker_cb_dnd_data, 67
 - ewl_colorpicker_cb_previous_clicked, 68
 - ewl_colorpicker_cb_radio_change, 68
 - ewl_colorpicker_cb_spinner_change, 68
 - ewl_colorpicker_cb_square_change, 68
 - ewl_colorpicker_cb_vertical_change, 68
 - ewl_colorpicker_color_mode_get, 68
 - ewl_colorpicker_color_mode_set, 68
 - ewl_colorpicker_current_rgb_get, 68
 - ewl_colorpicker_current_rgb_set, 69
 - ewl_colorpicker_has_alpha_get, 69
 - ewl_colorpicker_has_alpha_set, 69
 - ewl_colorpicker_init, 69
 - EWL_COLORPICKER_IS, 67
 - ewl_colorpicker_new, 70
 - ewl_colorpicker_previous_rgb_get, 70
 - ewl_colorpicker_previous_rgb_set, 70
 - EWL_COLORPICKER_TYPE, 67

- g, 604
- h, 604
- has_alpha, 604
- hsv, 604
- mode, 604
- picker, 605
- preview, 605
- previous, 605
- r, 605
- rgb, 605
- s, 605
- spinners, 605
- square, 605
- updating, 605
- v, 605
- vertical, 605
- Ewl_Colorpicker: The colour picking widget, 65
- ewl_colorpicker_alpha_get
 - Ewl_Colorpicker, 67
- ewl_colorpicker_alpha_set
 - Ewl_Colorpicker, 67
- ewl_colorpicker_cb_dnd_data
 - Ewl_Colorpicker, 67
 - lib/ewl_colorpicker.c, 868
- ewl_colorpicker_cb_previous_clicked
 - Ewl_Colorpicker, 68
 - lib/ewl_colorpicker.c, 868
- ewl_colorpicker_cb_radio_change
 - Ewl_Colorpicker, 68
 - lib/ewl_colorpicker.c, 868
- ewl_colorpicker_cb_spinner_change
 - Ewl_Colorpicker, 68
 - lib/ewl_colorpicker.c, 868
- ewl_colorpicker_cb_square_change
 - Ewl_Colorpicker, 68
 - lib/ewl_colorpicker.c, 868
- ewl_colorpicker_cb_vertical_change
 - Ewl_Colorpicker, 68
 - lib/ewl_colorpicker.c, 868
- ewl_colorpicker_color_mode_get
 - Ewl_Colorpicker, 68
- ewl_colorpicker_color_mode_set
 - Ewl_Colorpicker, 68
- ewl_colorpicker_current_rgb_get
 - Ewl_Colorpicker, 68
- ewl_colorpicker_current_rgb_set
 - Ewl_Colorpicker, 69
- ewl_colorpicker_has_alpha_get
 - Ewl_Colorpicker, 69
- ewl_colorpicker_has_alpha_set
 - Ewl_Colorpicker, 69
- ewl_colorpicker_init
 - Ewl_Colorpicker, 69
- EWL_COLORPICKER_IS
 - Ewl_Colorpicker, 67
- ewl_colorpicker_new
 - Ewl_Colorpicker, 70
- ewl_colorpicker_previous_rgb_get
 - Ewl_Colorpicker, 70
- ewl_colorpicker_previous_rgb_set
 - Ewl_Colorpicker, 70
- EWL_COLORPICKER_RADIOBUTTON
 - lib/ewl_colorpicker.c, 867
- Ewl_Colorpicker_Radiobutton, 607
 - button, 607
 - lib/ewl_colorpicker.c, 867
 - mode, 607
- EWL_COLORPICKER_RADIOBUTTON_ -
 - TYPE
 - lib/ewl_colorpicker.c, 867
- EWL_COLORPICKER_SPINNER
 - lib/ewl_colorpicker.c, 867
- Ewl_Colorpicker_Spinner, 609
 - lib/ewl_colorpicker.c, 867
 - mode, 609
 - spinner, 609
- EWL_COLORPICKER_SPINNER_TYPE
 - lib/ewl_colorpicker.c, 867
- EWL_COLORPICKER_TYPE
 - Ewl_Colorpicker, 67
- EWL_COMBO
 - Ewl_Combo, 73
- Ewl_Combo, 610
 - button, 610
 - editable, 610
- EWL_COMBO, 73
- Ewl_Combo, 73
 - ewl_combo_cb_decrement_clicked, 74
 - ewl_combo_cb_popup_hide, 74
- EWL_COMBO_CELL, 73
- Ewl_Combo_Cell, 73
 - ewl_combo_cell_combo_get, 74
 - ewl_combo_cell_combo_set, 74
 - ewl_combo_cell_data_get, 74
 - ewl_combo_cell_data_set, 74
 - ewl_combo_cell_init, 75
- EWL_COMBO_CELL_IS, 73
 - ewl_combo_cell_model_get, 75
 - ewl_combo_cell_model_set, 75
 - ewl_combo_cell_new, 75
- EWL_COMBO_CELL_TYPE, 73
 - ewl_combo_editable_get, 75
 - ewl_combo_editable_set, 76
 - ewl_combo_init, 76
- EWL_COMBO_IS, 73
 - ewl_combo_new, 76
 - ewl_combo_popup_container_set, 76
 - ewl_combo_scrollable_get, 77

- ewl_combo_scrollable_set, 77
- EWL_COMBO_TYPE, 73
- header, 610
- mvc, 611
- popup, 611
- scrollable, 611
- ewl_combo.h
 - ewl_combo_cell_cb_clicked, 876
- Ewl_Combo: A Simple Combo Box, 71
- ewl_combo_cb_decrement_clicked
 - Ewl_Combo, 74
 - lib/ewl_combo.c, 873
- ewl_combo_cb_popup_hide
 - Ewl_Combo, 74
 - lib/ewl_combo.c, 873
- EWL_COMBO_CELL
 - Ewl_Combo, 73
- Ewl_Combo_Cell, 612
 - cell, 612
 - combo, 612
 - Ewl_Combo, 73
 - model, 612
 - mvc_data, 613
- ewl_combo_cell_cb_clicked
 - ewl_combo.h, 876
 - lib/ewl_combo.c, 873
- ewl_combo_cell_combo_get
 - Ewl_Combo, 74
- ewl_combo_cell_combo_set
 - Ewl_Combo, 74
- ewl_combo_cell_data_get
 - Ewl_Combo, 74
- ewl_combo_cell_data_set
 - Ewl_Combo, 74
- ewl_combo_cell_init
 - Ewl_Combo, 75
- EWL_COMBO_CELL_IS
 - Ewl_Combo, 73
- ewl_combo_cell_model_get
 - Ewl_Combo, 75
- ewl_combo_cell_model_set
 - Ewl_Combo, 75
- ewl_combo_cell_new
 - Ewl_Combo, 75
- EWL_COMBO_CELL_TYPE
 - Ewl_Combo, 73
- ewl_combo_editable_get
 - Ewl_Combo, 75
- ewl_combo_editable_set
 - Ewl_Combo, 76
- ewl_combo_init
 - Ewl_Combo, 76
- EWL_COMBO_IS
 - Ewl_Combo, 73
- ewl_combo_new
 - Ewl_Combo, 76
- ewl_combo_popup_container_set
 - Ewl_Combo, 76
- ewl_combo_scrollable_get
 - Ewl_Combo, 77
- ewl_combo_scrollable_set
 - Ewl_Combo, 77
- ewl_combo_submenu_new
 - lib/ewl_combo.c, 873
- EWL_COMBO_TYPE
 - Ewl_Combo, 73
- Ewl_Config, 614
 - app_name, 614
 - data, 614
 - Ewl_Config, 79
 - ewl_config_cache_init, 79
 - ewl_config_can_save_system, 79
 - ewl_config_color_get, 79
 - ewl_config_color_set, 80
 - ewl_config_destroy, 80
 - ewl_config_float_get, 80
 - ewl_config_float_set, 80
 - ewl_config_init, 81
 - ewl_config_int_get, 81
 - ewl_config_int_set, 81
 - ewl_config_new, 81
 - ewl_config_shutdown, 82
 - ewl_config_string_get, 82
 - ewl_config_string_set, 82
 - ewl_config_system_save, 82
 - ewl_config_user_save, 82
 - instance, 614
 - system, 614
 - user, 614
- ewl_config
 - ewl_config.c, 878
 - ewl_private.h, 1153
- ewl_config.c
 - ewl_config, 878
 - ewl_config_cache, 878
 - ewl_embed_list, 878
- Ewl_Config: Functions for Manipulating Configuration Data, 78
- Ewl_Config_Cache, 615
 - backtrace, 615
 - enable, 615
 - evas_render, 615
 - ewl_debug.h, 904
 - gc_reap, 615
 - level, 615
 - print_keys, 615
 - print_signals, 615
 - segv, 616

- ewl_config_cache
 - ewl_config.c, 878
 - ewl_debug.h, 905
- EWL_CONFIG_CACHE_EVAS_FONT
 - ewl_private.h, 1153
- EWL_CONFIG_CACHE_EVAS_IMAGE
 - ewl_private.h, 1153
- ewl_config_cache_init
 - Ewl_Config, 79
- ewl_config_can_save_system
 - Ewl_Config, 79
- ewl_config_color_get
 - Ewl_Config, 79
- ewl_config_color_set
 - Ewl_Config, 80
- EWL_CONFIG_DEBUG_BACKTRACE
 - ewl_private.h, 1153
- EWL_CONFIG_DEBUG_ENABLE
 - ewl_private.h, 1153
- EWL_CONFIG_DEBUG_EVAS_RENDER
 - ewl_private.h, 1153
- EWL_CONFIG_DEBUG_GC_REAP
 - ewl_private.h, 1153
- EWL_CONFIG_DEBUG_LEVEL
 - ewl_private.h, 1153
- EWL_CONFIG_DEBUG_SEGV
 - ewl_private.h, 1153
- ewl_config_destroy
 - Ewl_Config, 80
- EWL_CONFIG_ENGINE_NAME
 - ewl_private.h, 1153
- ewl_config_float_get
 - Ewl_Config, 80
- ewl_config_float_set
 - Ewl_Config, 80
- ewl_config_init
 - Ewl_Config, 81
- ewl_config_int_get
 - Ewl_Config, 81
- ewl_config_int_set
 - Ewl_Config, 81
- ewl_config_new
 - Ewl_Config, 81
- ewl_config_shutdown
 - Ewl_Config, 82
- ewl_config_string_get
 - Ewl_Config, 82
- ewl_config_string_set
 - Ewl_Config, 82
- ewl_config_system_save
 - Ewl_Config, 82
- EWL_CONFIG_THEME_COLOR_-
 - CLASSES_COUNT
 - ewl_private.h, 1153
- EWL_CONFIG_THEME_COLOR_-
 - CLASSES_OVERRIDE
 - ewl_private.h, 1153
- EWL_CONFIG_THEME_ICON_SIZE
 - ewl_private.h, 1153
- EWL_CONFIG_THEME_ICON_THEME
 - ewl_private.h, 1153
- EWL_CONFIG_THEME_NAME
 - ewl_private.h, 1153
- EWL_CONFIG_THEME_PRINT_KEYS
 - ewl_private.h, 1153
- EWL_CONFIG_THEME_PRINT_-
 - SIGNALS
 - ewl_private.h, 1153
- ewl_config_user_save
 - Ewl_Config, 82
- Ewl_Configure_Queue, 617
 - buffer, 617
 - end, 617
 - ewl_misc.c, 1101
- EWL_CONFIGURE_QUEUE_SIZE
 - ewl_misc.c, 1101
- ewl_configure_request
 - Ewl_Misc, 313
- EWL_CONTAINER
 - Ewl_Container, 87
- Ewl_Container, 618
 - child_add, 618
 - child_hide, 618
 - child_remove, 618
 - child_resize, 618
 - child_show, 619
 - children, 619
 - clip_box, 619
 - Ewl_Child_Add, 88
 - Ewl_Child_Hide, 88
 - Ewl_Child_Remove, 88
 - Ewl_Child_Resize, 88
 - Ewl_Child_Show, 88
 - EWL_CONTAINER, 87
 - Ewl_Container, 88
 - ewl_container_add_notify_set, 88
 - ewl_container_callback_intercept, 89
 - ewl_container_callback_nointercept, 89
 - ewl_container_callback_notify, 89
 - ewl_container_cb_configure, 90
 - ewl_container_cb_disable, 90
 - ewl_container_cb_enable, 90
 - ewl_container_cb_obscure, 90
 - ewl_container_cb_realize, 90
 - ewl_container_cb_reparent, 90
 - ewl_container_cb_reveal, 90
 - ewl_container_cb_unrealize, 90
 - ewl_container_child_add_call, 90

- ewl_container_child_append, 90
- ewl_container_child_at_get, 91
- ewl_container_child_at_recursive_get, 91
- ewl_container_child_count_get, 91
- ewl_container_child_count_internal_get, 91
- ewl_container_child_get, 92
- ewl_container_child_hide_call, 92
- ewl_container_child_index_get, 92
- ewl_container_child_index_internal_get, 92
- ewl_container_child_insert, 93
- ewl_container_child_insert_internal, 93
- ewl_container_child_internal_get, 93
- ewl_container_child_iterate_begin, 94
- ewl_container_child_iterator_set, 94
- ewl_container_child_next, 94
- ewl_container_child_prepend, 94
- ewl_container_child_remove, 95
- ewl_container_child_remove_call, 95
- ewl_container_child_resize, 95
- ewl_container_child_show_call, 96
- ewl_container_child_destroy, 96
- ewl_container_end_redirect_get, 96
- ewl_container_hide_notify_set, 96
- ewl_container_init, 97
- EWL_CONTAINER_IS, 87
- Ewl_Container_Iterator, 88
- ewl_container_largest_prefer, 97
- ewl_container_redirect_get, 97
- ewl_container_redirect_set, 97
- ewl_container_remove_notify_set, 98
- ewl_container_reset, 98
- ewl_container_resize_notify_set, 98
- ewl_container_show_notify_set, 99
- ewl_container_sum_prefer, 99
- EWL_CONTAINER_TYPE, 87
- iterator, 619
- redirect, 619
- widget, 619
- Ewl_Container: Widgets Holding Other Widgets, 84
- ewl_container_add_notify_set
 - Ewl_Container, 88
- ewl_container_callback_intercept
 - Ewl_Container, 89
- ewl_container_callback_nointercept
 - Ewl_Container, 89
- ewl_container_callback_notify
 - Ewl_Container, 89
- ewl_container_cb_configure
 - Ewl_Container, 90
 - lib/ewl_container.c, 884
- ewl_container_cb_disable
 - Ewl_Container, 90
 - lib/ewl_container.c, 884
- ewl_container_cb_enable
 - Ewl_Container, 90
 - lib/ewl_container.c, 884
- ewl_container_cb_obscure
 - Ewl_Container, 90
 - lib/ewl_container.c, 884
- ewl_container_cb_realize
 - Ewl_Container, 90
 - lib/ewl_container.c, 884
- ewl_container_cb_reparent
 - Ewl_Container, 90
 - lib/ewl_container.c, 884
- ewl_container_cb_reveal
 - Ewl_Container, 90
 - lib/ewl_container.c, 884
- ewl_container_cb_unrealize
 - Ewl_Container, 90
 - lib/ewl_container.c, 884
- ewl_container_child_add_call
 - Ewl_Container, 90
- ewl_container_child_append
 - Ewl_Container, 90
- ewl_container_child_at_get
 - Ewl_Container, 91
- ewl_container_child_at_recursive_get
 - Ewl_Container, 91
- ewl_container_child_count_get
 - Ewl_Container, 91
- ewl_container_child_count_internal_get
 - Ewl_Container, 91
- ewl_container_child_get
 - Ewl_Container, 92
- ewl_container_child_hide_call
 - Ewl_Container, 92
- ewl_container_child_index_get
 - Ewl_Container, 92
- ewl_container_child_index_internal_get
 - Ewl_Container, 92
- ewl_container_child_insert
 - Ewl_Container, 93
- ewl_container_child_insert_internal
 - Ewl_Container, 93
- ewl_container_child_internal_get
 - Ewl_Container, 93
- ewl_container_child_iterate_begin
 - Ewl_Container, 94
- ewl_container_child_iterator_set
 - Ewl_Container, 94
- ewl_container_child_next
 - Ewl_Container, 94
- ewl_container_child_prepend

- Ewl_Container, 94
- ewl_container_child_remove
 - Ewl_Container, 95
- ewl_container_child_remove_call
 - Ewl_Container, 95
- ewl_container_child_resize
 - Ewl_Container, 95
- ewl_container_child_show_call
 - Ewl_Container, 96
- ewl_container_destroy
 - Ewl_Container, 96
- ewl_container_end_redirect_get
 - Ewl_Container, 96
- ewl_container_hide_notify_set
 - Ewl_Container, 96
- ewl_container_init
 - Ewl_Container, 97
- EWL_CONTAINER_IS
 - Ewl_Container, 87
- Ewl_Container_Iterator
 - Ewl_Container, 88
- ewl_container_largest_prefer
 - Ewl_Container, 97
- ewl_container_redirect_get
 - Ewl_Container, 97
- ewl_container_redirect_set
 - Ewl_Container, 97
- ewl_container_remove_notify_set
 - Ewl_Container, 98
- ewl_container_reset
 - Ewl_Container, 98
- ewl_container_resize_notify_set
 - Ewl_Container, 98
- ewl_container_show_notify_set
 - Ewl_Container, 99
- ewl_container_sum_prefer
 - Ewl_Container, 99
- EWL_CONTAINER_TYPE
 - Ewl_Container, 87
- EWL_CONTEXT_MENU
 - Ewl_Context_Menu, 101
- Ewl_Context_Menu, 620
 - child_add, 621
 - child_remove, 621
 - container, 621
 - EWL_CONTEXT_MENU, 101
 - Ewl_Context_Menu, 101
 - ewl_context_menu_attach, 101
 - ewl_context_menu_cb_attach_mouse_down, 101
 - ewl_context_menu_cb_child_add, 102
 - ewl_context_menu_cb_child_clicked, 102
- ewl_context_menu_cb_child_mouse_in, 102
- ewl_context_menu_cb_child_remove, 102
- ewl_context_menu_cb_focus_in, 102
- ewl_context_menu_cb_hide, 102
- ewl_context_menu_cb_mouse_down, 102
- ewl_context_menu_cb_mouse_move, 102
- ewl_context_menu_container_set, 102
- ewl_context_menu_detach, 102
- ewl_context_menu_init, 102
- EWL_CONTEXT_MENU_IS, 101
- ewl_context_menu_new, 103
- EWL_CONTEXT_MENU_TYPE, 101
- open_menu, 621
- popup, 621
- ewl_context_menu.c
 - ewl_context_menu_cb_attach_mouse_down, 891
 - ewl_context_menu_cb_child_clicked, 891
 - ewl_context_menu_cb_child_mouse_in, 891
 - ewl_context_menu_cb_focus_in, 891
 - ewl_context_menu_cb_hide, 891
 - ewl_context_menu_cb_mouse_down, 891
 - ewl_context_menu_cb_mouse_move, 891
- Ewl_Context_Menu: A context menu, 100
- ewl_context_menu_attach
 - Ewl_Context_Menu, 101
- ewl_context_menu_cb_attach_mouse_down
 - Ewl_Context_Menu, 101
 - ewl_context_menu.c, 891
- ewl_context_menu_cb_child_add
 - Ewl_Context_Menu, 102
- ewl_context_menu_cb_child_clicked
 - Ewl_Context_Menu, 102
 - ewl_context_menu.c, 891
- ewl_context_menu_cb_child_mouse_in
 - Ewl_Context_Menu, 102
 - ewl_context_menu.c, 891
- ewl_context_menu_cb_child_remove
 - Ewl_Context_Menu, 102
- ewl_context_menu_cb_focus_in
 - Ewl_Context_Menu, 102
 - ewl_context_menu.c, 891
- ewl_context_menu_cb_hide
 - Ewl_Context_Menu, 102
 - ewl_context_menu.c, 891
- ewl_context_menu_cb_mouse_down

- Ewl_Context_Menu, 102
- ewl_context_menu.c, 891
- ewl_context_menu_cb_mouse_move
 - Ewl_Context_Menu, 102
 - ewl_context_menu.c, 891
- ewl_context_menu_container_set
 - Ewl_Context_Menu, 102
- ewl_context_menu_detach
 - Ewl_Context_Menu, 102
- ewl_context_menu_init
 - Ewl_Context_Menu, 102
- EWL_CONTEXT_MENU_IS
 - Ewl_Context_Menu, 101
- ewl_context_menu_new
 - Ewl_Context_Menu, 103
- EWL_CONTEXT_MENU_TYPE
 - Ewl_Context_Menu, 101
- EWL_CURSOR
 - Ewl_Cursor, 104
- Ewl_Cursor, 622
 - EWL_CURSOR, 104
 - Ewl_Cursor, 105
 - ewl_cursor_init, 105
 - EWL_CURSOR_IS, 104
 - ewl_cursor_new, 105
 - EWL_CURSOR_TYPE, 104
 - handle, 622
 - refcount, 622
 - window, 623
- Ewl_Cursor: Container to create custom mouse
 - cursors, 104
- ewl_cursor_init
 - Ewl_Cursor, 105
- EWL_CURSOR_IS
 - Ewl_Cursor, 104
- ewl_cursor_new
 - Ewl_Cursor, 105
- EWL_CURSOR_TYPE
 - Ewl_Cursor, 104
- EWL_DATEPICKER
 - Ewl_Datepicker, 106
- Ewl_Datepicker, 624
 - calendar, 624
 - calendar_window, 624
 - EWL_DATEPICKER, 106
 - Ewl_Datepicker, 107
 - ewl_datepicker_cb_destroy, 107
 - ewl_datepicker_cb_value_changed, 107
 - ewl_datepicker_cb_window_mouse_down, 107
 - ewl_datepicker_init, 107
 - EWL_DATEPICKER_IS, 106
 - ewl_datepicker_new, 107
 - EWL_DATEPICKER_TYPE, 107
 - text, 624
- Ewl_Datepicker: The EWL Datepicker widget, 106
- ewl_datepicker_cb_destroy
 - Ewl_Datepicker, 107
 - lib/ewl_datepicker.c, 897
- ewl_datepicker_cb_value_changed
 - Ewl_Datepicker, 107
 - lib/ewl_datepicker.c, 897
- ewl_datepicker_cb_window_mouse_down
 - Ewl_Datepicker, 107
 - lib/ewl_datepicker.c, 897
- ewl_datepicker_init
 - Ewl_Datepicker, 107
- EWL_DATEPICKER_IS
 - Ewl_Datepicker, 106
- ewl_datepicker_new
 - Ewl_Datepicker, 107
- EWL_DATEPICKER_TYPE
 - Ewl_Datepicker, 107
- ewl_debug.h
 - DCHECK_PARAM_PTR, 903
 - DCHECK_PARAM_PTR_RET, 903
 - DCHECK_TYPE, 903
 - DCHECK_TYPE_RET, 903
 - DENTER_FUNCTION, 903
 - DERROR, 903
 - DLEAVE_FUNCTION, 903
 - DLEVEL_STABLE, 904
 - DLEVEL_TESTING, 904
 - DLEVEL_UNSTABLE, 904
 - DRETURN, 904
 - DRETURN_FLOAT, 904
 - DRETURN_INT, 904
 - DRETURN_PTR, 904
 - DWARNING, 904
 - ewl_backtrace, 904
 - Ewl_Config_Cache, 904
 - ewl_config_cache, 905
 - ewl_print_warning, 904
 - ewl_segv, 904
- ewl_debug_indent_print
 - Ewl_Misc, 313
- ewl_destroy_request
 - Ewl_Misc, 313
- EWL_DIALOG
 - Ewl_Dialog, 109
- Ewl_Dialog, 626
 - action_area, 627
 - active_area, 627
 - box, 627
 - EWL_DIALOG, 109
 - Ewl_Dialog, 109
 - ewl_dialog_action_position_get, 109

- ewl_dialog_action_position_set, 109
- ewl_dialog_active_area_get, 110
- ewl_dialog_active_area_set, 110
- ewl_dialog_has_separator_get, 110
- ewl_dialog_has_separator_set, 110
- ewl_dialog_init, 110
- EWL_DIALOG_IS, 109
- ewl_dialog_new, 111
- EWL_DIALOG_TYPE, 109
- position, 627
- separator, 627
- vbox, 627
- window, 627
- Ewl_Dialog: A dialog window, 108
- ewl_dialog_action_position_get
 - Ewl_Dialog, 109
- ewl_dialog_action_position_set
 - Ewl_Dialog, 109
- ewl_dialog_active_area_get
 - Ewl_Dialog, 110
- ewl_dialog_active_area_set
 - Ewl_Dialog, 110
- ewl_dialog_has_separator_get
 - Ewl_Dialog, 110
- ewl_dialog_has_separator_set
 - Ewl_Dialog, 110
- ewl_dialog_init
 - Ewl_Dialog, 110
- EWL_DIALOG_IS
 - Ewl_Dialog, 109
- ewl_dialog_new
 - Ewl_Dialog, 111
- EWL_DIALOG_TYPE
 - Ewl_Dialog, 109
- Ewl_Dnd
 - EWL_CALLBACK_DND_DATA_RECEIVED, 117
 - EWL_CALLBACK_DND_DATA_REQUEST, 117
 - EWL_CALLBACK_DND_DROP, 117
 - EWL_CALLBACK_DND_ENTER, 117
 - EWL_CALLBACK_DND_LEAVE, 117
 - EWL_CALLBACK_DND_POSITION, 117
 - ewl_dnd_accepted_types_contains, 114
 - ewl_dnd_accepted_types_get, 114
 - ewl_dnd_accepted_types_set, 114
 - ewl_dnd_disable, 115
 - ewl_dnd_drag_drop, 115
 - ewl_dnd_drag_start, 115
 - ewl_dnd_drag_widget_clear, 115
 - ewl_dnd_drag_widget_get, 115
 - ewl_dnd_enable, 116
 - ewl_dnd_init, 116
 - ewl_dnd_provided_types_contains, 116
 - ewl_dnd_provided_types_get, 116
 - ewl_dnd_provided_types_set, 116
 - ewl_dnd_shutdown, 117
 - ewl_dnd_status_get, 117
- ewl_dnd.c
 - EWL_DND_WINDOW_ROOT, 912
- Ewl_Dnd: The files containing DND functions, 112
- ewl_dnd_accepted_types_contains
 - Ewl_Dnd, 114
- ewl_dnd_accepted_types_get
 - Ewl_Dnd, 114
- ewl_dnd_accepted_types_set
 - Ewl_Dnd, 114
- ewl_dnd_client_message_handler
 - ewl_dnd_snoop.c, 807
- ewl_dnd_disable
 - Ewl_Dnd, 115
- ewl_dnd_drag_drop
 - Ewl_Dnd, 115
- ewl_dnd_drag_start
 - Ewl_Dnd, 115
- ewl_dnd_drag_widget_clear
 - Ewl_Dnd, 115
- ewl_dnd_drag_widget_get
 - Ewl_Dnd, 115
- ewl_dnd_drop_handler
 - ewl_dnd_snoop.c, 807
- ewl_dnd_enable
 - Ewl_Dnd, 116
- ewl_dnd_enter_handler
 - ewl_dnd_snoop.c, 807
- ewl_dnd_finished_handler
 - ewl_dnd_snoop.c, 807
- ewl_dnd_init
 - Ewl_Dnd, 116
- ewl_dnd_leave_handler
 - ewl_dnd_snoop.c, 807
- ewl_dnd_position_handler
 - ewl_dnd_snoop.c, 807
- ewl_dnd_provided_types_contains
 - Ewl_Dnd, 116
- ewl_dnd_provided_types_get
 - Ewl_Dnd, 116
- ewl_dnd_provided_types_set
 - Ewl_Dnd, 116
- ewl_dnd_selection_clear_handler
 - ewl_dnd_snoop.c, 807
- ewl_dnd_selection_notify_handler
 - ewl_dnd_snoop.c, 807
- ewl_dnd_selection_request_handler
 - ewl_dnd_snoop.c, 807
- ewl_dnd_shutdown

- Ewl_Dnd, 117
- ewl_dnd_snoop.c
 - ewl_dnd_client_message_handler, 807
 - ewl_dnd_drop_handler, 807
 - ewl_dnd_enter_handler, 807
 - ewl_dnd_finished_handler, 807
 - ewl_dnd_leave_handler, 807
 - ewl_dnd_position_handler, 807
 - ewl_dnd_selection_clear_handler, 807
 - ewl_dnd_selection_notify_handler, 807
 - ewl_dnd_selection_request_handler, 807
 - ewl_dnd_status_handler, 807
 - test_info, 807
- ewl_dnd_status_get
 - Ewl_Dnd, 117
- ewl_dnd_status_handler
 - ewl_dnd_snoop.c, 807
- Ewl_Dnd_Types, 628
 - Ewl_Events, 167
 - num_types, 628
 - types, 628
- EWL_DND_WINDOW_ROOT
 - ewl_dnd.c, 912
- EWL_EMBED
 - Ewl_Embed, 122
- Ewl_Embed, 629
 - canvas, 630
 - canvas_window, 630
 - clicked, 630
 - cursor, 630
 - dnd_count, 630
 - dnd_last_position, 630
 - dnd_types, 630
 - drag_widget, 630
 - drop_widget, 630
 - engine, 630
 - engine_name, 631
 - ev_clip, 631
 - EWL_EMBED, 122
 - Ewl_Embed, 122
 - ewl_embed_active_embed_get, 123
 - ewl_embed_active_set, 123
 - ewl_embed_cache_cleanup, 123
 - ewl_embed_canvas_set, 123
 - ewl_embed_canvas_window_find, 123
 - ewl_embed_cb_configure, 124
 - ewl_embed_cb_destroy, 124
 - ewl_embed_cb_focus_out, 124
 - ewl_embed_cb_realize, 124
 - ewl_embed_cb_unrealize, 124
 - ewl_embed_coord_to_screen, 124
 - ewl_embed_desktop_size_get, 124
 - ewl_embed_dnd_aware_remove, 125
 - ewl_embed_dnd_aware_set, 125
 - ewl_embed_dnd_data_received_feed, 125
 - ewl_embed_dnd_data_request_feed, 125
 - ewl_embed_dnd_drop_feed, 126
 - ewl_embed_dnd_position_feed, 126
 - ewl_embed_engine_name_get, 126
 - ewl_embed_engine_name_set, 127
 - ewl_embed_focus_get, 127
 - ewl_embed_focus_set, 127
 - ewl_embed_focused_widget_get, 127
 - ewl_embed_focused_widget_set, 128
 - ewl_embed_font_path_add, 128
 - ewl_embed_freeze, 128
 - ewl_embed_info_widgets_cleanup, 128
 - ewl_embed_init, 129
 - EWL_EMBED_IS, 122
 - ewl_embed_key_down_feed, 129
 - ewl_embed_key_up_feed, 129
 - ewl_embed_mouse_cursor_set, 130
 - ewl_embed_mouse_down_feed, 130
 - ewl_embed_mouse_move_feed, 130
 - ewl_embed_mouse_out_feed, 130
 - ewl_embed_mouse_up_feed, 131
 - ewl_embed_mouse_wheel_feed, 131
 - ewl_embed_new, 131
 - ewl_embed_object_cache, 132
 - ewl_embed_object_request, 132
 - ewl_embed_position_get, 132
 - ewl_embed_shutdown, 132
 - ewl_embed_tab_order_append, 133
 - ewl_embed_tab_order_insert, 133
 - ewl_embed_tab_order_insert_after, 133
 - ewl_embed_tab_order_insert_before, 133
 - ewl_embed_tab_order_next, 134
 - ewl_embed_tab_order_prepend, 134
 - ewl_embed_tab_order_previous, 134
 - ewl_embed_tab_order_remove, 134
 - ewl_embed_thaw, 135
 - EWL_EMBED_TYPE, 122
 - ewl_embed_widget_find, 135
 - EWL_EMBED_WINDOW, 122
 - Ewl_Embed_Window, 122
 - ewl_embed_window_position_get, 135
 - focus, 631
 - focused, 631
 - last, 631
 - mouse_in, 631
 - obj_cache, 631
 - overlay, 631
 - smart, 631
 - tab_order, 631
 - x, 631
 - y, 631

- ewl_embed.c
 - ewl_embed_cb_configure, 919
 - ewl_embed_cb_destroy, 919
 - ewl_embed_cb_focus_out, 919
 - ewl_embed_cb_realize, 919
 - ewl_embed_cb_unrealize, 919
 - ewl_embed_list, 919
- Ewl_Embed: A Container for Displaying on an
 - canvas, 118
- ewl_embed_active_embed_get
 - Ewl_Embed, 123
- ewl_embed_active_set
 - Ewl_Embed, 123
- ewl_embed_cache_cleanup
 - Ewl_Embed, 123
- ewl_embed_canvas_set
 - Ewl_Embed, 123
- ewl_embed_canvas_window_find
 - Ewl_Embed, 123
- ewl_embed_cb_configure
 - Ewl_Embed, 124
 - ewl_embed.c, 919
- ewl_embed_cb_destroy
 - Ewl_Embed, 124
 - ewl_embed.c, 919
- ewl_embed_cb_focus_out
 - Ewl_Embed, 124
 - ewl_embed.c, 919
- ewl_embed_cb_realize
 - Ewl_Embed, 124
 - ewl_embed.c, 919
- ewl_embed_cb_unrealize
 - Ewl_Embed, 124
 - ewl_embed.c, 919
- ewl_embed_coord_to_screen
 - Ewl_Embed, 124
- ewl_embed_desktop_size_get
 - Ewl_Embed, 124
- ewl_embed_dnd_aware_remove
 - Ewl_Embed, 125
- ewl_embed_dnd_aware_set
 - Ewl_Embed, 125
- ewl_embed_dnd_data_received_feed
 - Ewl_Embed, 125
- ewl_embed_dnd_data_request_feed
 - Ewl_Embed, 125
- ewl_embed_dnd_drop_feed
 - Ewl_Embed, 126
- ewl_embed_dnd_position_feed
 - Ewl_Embed, 126
- ewl_embed_engine_name_get
 - Ewl_Embed, 126
- ewl_embed_engine_name_set
 - Ewl_Embed, 127
- ewl_embed_focus_get
 - Ewl_Embed, 127
- ewl_embed_focus_set
 - Ewl_Embed, 127
- ewl_embed_focused_widget_get
 - Ewl_Embed, 127
- ewl_embed_focused_widget_set
 - Ewl_Embed, 128
- ewl_embed_font_path_add
 - Ewl_Embed, 128
- ewl_embed_freeze
 - Ewl_Embed, 128
- ewl_embed_info_widgets_cleanup
 - Ewl_Embed, 128
- ewl_embed_init
 - Ewl_Embed, 129
- EWL_EMBED_IS
 - Ewl_Embed, 122
- ewl_embed_key_down_feed
 - Ewl_Embed, 129
- ewl_embed_key_up_feed
 - Ewl_Embed, 129
- ewl_embed_list
 - ewl_config.c, 878
 - ewl_embed.c, 919
 - ewl_events.c, 977
 - ewl_misc.c, 1101
 - lib/ewl_theme.c, 1235
- ewl_embed_mouse_cursor_set
 - Ewl_Embed, 130
- ewl_embed_mouse_down_feed
 - Ewl_Embed, 130
- ewl_embed_mouse_move_feed
 - Ewl_Embed, 130
- ewl_embed_mouse_out_feed
 - Ewl_Embed, 130
- ewl_embed_mouse_up_feed
 - Ewl_Embed, 131
- ewl_embed_mouse_wheel_feed
 - Ewl_Embed, 131
- ewl_embed_new
 - Ewl_Embed, 131
- ewl_embed_object_cache
 - Ewl_Embed, 132
- ewl_embed_object_request
 - Ewl_Embed, 132
- ewl_embed_position_get
 - Ewl_Embed, 132
- ewl_embed_shutdown
 - Ewl_Embed, 132
- ewl_embed_tab_order_append
 - Ewl_Embed, 133
- ewl_embed_tab_order_insert
 - Ewl_Embed, 133

- ewl_embed_tab_order_insert_after
 - Ewl_Embed, 133
- ewl_embed_tab_order_insert_before
 - Ewl_Embed, 133
- ewl_embed_tab_order_next
 - Ewl_Embed, 134
- ewl_embed_tab_order_prepend
 - Ewl_Embed, 134
- ewl_embed_tab_order_previous
 - Ewl_Embed, 134
- ewl_embed_tab_order_remove
 - Ewl_Embed, 134
- ewl_embed_test.c
 - __UNUSED__, 796
 - edje, 796
 - ee, 796
 - entry, 796
 - evas, 796
 - main, 796
 - text, 796
- ewl_embed_thaw
 - Ewl_Embed, 135
- EWL_EMBED_TYPE
 - Ewl_Embed, 122
- ewl_embed_widget_find
 - Ewl_Embed, 135
- EWL_EMBED_WINDOW
 - Ewl_Embed, 122
- Ewl_Embed_Window
 - Ewl_Embed, 122
- ewl_embed_window_position_get
 - Ewl_Embed, 135
- EWL_ENGINE
 - ewl_engines.h, 947
- Ewl_Engine, 633
 - dependencies, 633
 - ewl_engines.h, 947
 - functions, 633
 - handle, 633
 - name, 633
- ewl_engine.c
 - test_info, 808
 - WIN_NAME, 808
- EWL_ENGINE_CANVAS_DAMAGE_ADD
 - ewl_engines.h, 952
- ewl_engine_canvas_damage_add
 - ewl_engines.c, 928
 - ewl_engines.h, 954
- EWL_ENGINE_CANVAS_FREEZE
 - ewl_engines.h, 952
- ewl_engine_canvas_freeze
 - ewl_engines.c, 928
 - ewl_engines.h, 954
- Ewl_Engine_Canvas_Hooks
 - ewl_engines.h, 947, 952
- EWL_ENGINE_CANVAS_MAX
 - ewl_engines.h, 952
- EWL_ENGINE_CANVAS_OUTPUT_SET
 - ewl_engines.h, 952
- ewl_engine_canvas_output_set
 - ewl_engines.c, 928
 - ewl_engines.h, 954
- EWL_ENGINE_CANVAS_RENDER
 - ewl_engines.h, 952
- ewl_engine_canvas_render
 - ewl_engines.c, 929
 - ewl_engines.h, 955
- EWL_ENGINE_CANVAS_SETUP
 - ewl_engines.h, 952
- ewl_engine_canvas_setup
 - ewl_engines.c, 929
 - ewl_engines.h, 955
- EWL_ENGINE_CANVAS_THAW
 - ewl_engines.h, 952
- ewl_engine_canvas_thaw
 - ewl_engines.c, 929
 - ewl_engines.h, 955
- Ewl_Engine_Cb_Canvas_Damage_Add
 - ewl_engines.h, 947
- Ewl_Engine_Cb_Canvas_Freeze
 - ewl_engines.h, 947
- Ewl_Engine_Cb_Canvas_Output_Set
 - ewl_engines.h, 947
- Ewl_Engine_Cb_Canvas_Render
 - ewl_engines.h, 947
- Ewl_Engine_Cb_Canvas_Setup
 - ewl_engines.h, 947
- Ewl_Engine_Cb_Canvas_Thaw
 - ewl_engines.h, 947
- Ewl_Engine_Cb_Keyboard_Grab
 - ewl_engines.h, 947
- Ewl_Engine_Cb_Keyboard_Ungrab
 - ewl_engines.h, 947
- Ewl_Engine_Cb_Pointer_Data_New
 - ewl_engines.h, 947
- Ewl_Engine_Cb_Pointer_Free
 - ewl_engines.h, 948
- Ewl_Engine_Cb_Pointer_Get
 - ewl_engines.h, 948
- Ewl_Engine_Cb_Pointer_Grab
 - ewl_engines.h, 948
- Ewl_Engine_Cb_Pointer_Set
 - ewl_engines.h, 948
- Ewl_Engine_Cb_Pointer_Ungrab
 - ewl_engines.h, 948
- Ewl_Engine_Cb_Theme_Clip_Add
 - ewl_engines.h, 948
- Ewl_Engine_Cb_Theme_Clip_Clipees_Get

- ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Clip_Color_Set
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Clip_Del
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Data_Get
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Freeze
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Add
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Clip_Set
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Clip_-
Unset
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Color_-
Class_Set
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Del
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_File_Set
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Hide
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Layer_-
Update
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Load_-
Error
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Max_-
Size_Get
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Min_-
Size_Get
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Move
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Part_-
Text_Set
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Resize
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Show
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Signal_-
Send
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Object_Stack_-
Add
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Thaw
ewl_engines.h, 949
- Ewl_Engine_Cb_Theme_Widget_Group
ewl_engines.h, 949
- Ewl_Engine_Cb_Window_Borderless_Set
ewl_engines.h, 949
- Ewl_Engine_Cb_Window_Desktop_Size_-
Get
ewl_engines.h, 949
- Ewl_Engine_Cb_Window_Destroy
ewl_engines.h, 950
- Ewl_Engine_Cb_Window_Dialog_Set
ewl_engines.h, 950
- Ewl_Engine_Cb_Window_Dnd_Aware_Set
ewl_engines.h, 950
- Ewl_Engine_Cb_Window_Dnd_Drag_-
Data_Send
ewl_engines.h, 950
- Ewl_Engine_Cb_Window_Dnd_Drag_Drop
ewl_engines.h, 950
- Ewl_Engine_Cb_Window_Dnd_Drag_Start
ewl_engines.h, 950
- Ewl_Engine_Cb_Window_Dnd_Drag_-
Types_Set
ewl_engines.h, 950
- Ewl_Engine_Cb_Window_Geometry_Get
ewl_engines.h, 950
- Ewl_Engine_Cb_Window_Hide
ewl_engines.h, 950
- Ewl_Engine_Cb_Window_Lower
ewl_engines.h, 951
- Ewl_Engine_Cb_Window_Min_Max_-
Size_Set
ewl_engines.h, 951
- Ewl_Engine_Cb_Window_Move
ewl_engines.h, 951
- Ewl_Engine_Cb_Window_Name_Class_Set
ewl_engines.h, 951
- Ewl_Engine_Cb_Window_New
ewl_engines.h, 951
- Ewl_Engine_Cb_Window_Raise
ewl_engines.h, 951
- Ewl_Engine_Cb_Window_Resize
ewl_engines.h, 951
- Ewl_Engine_Cb_Window_Selection_Text_-
Set
ewl_engines.h, 951
- Ewl_Engine_Cb_Window_Show
ewl_engines.h, 951
- Ewl_Engine_Cb_Window_States_Set
ewl_engines.h, 951
- Ewl_Engine_Cb_Window_Title_Set
ewl_engines.h, 951
- Ewl_Engine_Cb_Window_Transient_For
ewl_engines.h, 952
- ewl_engine_desktop_size_get

- ewl_engines.c, 929
- ewl_engines.h, 955
- EWL_ENGINE_DIR
 - ewl_engines.c, 928
- ewl_engine_embed_dnd_aware_set
 - ewl_engines.c, 930
 - ewl_engines.h, 956
- ewl_engine_embed_dnd_drag_data_send
 - ewl_engines.c, 930
 - ewl_engines.h, 956
- ewl_engine_embed_dnd_drag_drop
 - ewl_engines.c, 930
 - ewl_engines.h, 956
- ewl_engine_embed_dnd_drag_start
 - ewl_engines.c, 930
 - ewl_engines.h, 956
- ewl_engine_embed_dnd_drag_types_set
 - ewl_engines.c, 931
 - ewl_engines.h, 957
- ewl_engine_engine_shutdown
 - ewl_engines.h, 957
- ewl_engine_event_handlers_init
 - ewl_engines.h, 957
- ewl_engine_event_handlers_shutdown
 - ewl_engines.h, 957
- Ewl_Engine_Hook_Type
 - ewl_engines.c, 928
- EWL_ENGINE_HOOK_TYPE_CANVAS
 - ewl_engines.c, 928
- EWL_ENGINE_HOOK_TYPE_POINTER
 - ewl_engines.c, 928
- EWL_ENGINE_HOOK_TYPE_THEME
 - ewl_engines.c, 928
- EWL_ENGINE_HOOK_TYPE_WINDOW
 - ewl_engines.c, 928
- Ewl_Engine_Info, 634
 - canvas, 634
 - ewl_engines.h, 952
 - hooks, 634
 - pointer, 634
 - shutdown, 634
 - theme, 634
 - window, 634
- ewl_engine_keyboard_grab
 - ewl_engines.c, 931
 - ewl_engines.h, 957
- ewl_engine_keyboard_ungrab
 - ewl_engines.c, 931
 - ewl_engines.h, 957
- ewl_engine_names_get
 - ewl_engines.c, 931
 - ewl_engines.h, 957
- ewl_engine_new
 - ewl_engines.c, 932
- ewl_engines.h, 958
- EWL_ENGINE_POINTER_DATA_NEW
 - ewl_engines.h, 952
- ewl_engine_pointer_data_new
 - ewl_engines.c, 932
 - ewl_engines.h, 958
- EWL_ENGINE_POINTER_FREE
 - ewl_engines.h, 952
- ewl_engine_pointer_free
 - ewl_engines.c, 932
 - ewl_engines.h, 958
- EWL_ENGINE_POINTER_GET
 - ewl_engines.h, 952
- ewl_engine_pointer_get
 - ewl_engines.c, 932
 - ewl_engines.h, 958
- ewl_engine_pointer_grab
 - ewl_engines.c, 932
 - ewl_engines.h, 958
- Ewl_Engine_Pointer_Hooks
 - ewl_engines.h, 952
- EWL_ENGINE_POINTER_MAX
 - ewl_engines.h, 952
- EWL_ENGINE_POINTER_SET
 - ewl_engines.h, 952
- ewl_engine_pointer_set
 - ewl_engines.c, 933
 - ewl_engines.h, 959
- ewl_engine_pointer_ungrab
 - ewl_engines.c, 933
 - ewl_engines.h, 959
- EWL_ENGINE_THEME_CLIP_ADD
 - ewl_engines.h, 953
- EWL_ENGINE_THEME_CLIP_-
CLIP_EES_GET
 - ewl_engines.h, 953
- EWL_ENGINE_THEME_CLIP_COLOR_-
SET
 - ewl_engines.h, 953
- EWL_ENGINE_THEME_CLIP_DEL
 - ewl_engines.h, 953
- EWL_ENGINE_THEME_DATA_GET
 - ewl_engines.h, 953
- EWL_ENGINE_THEME_FREEZE
 - ewl_engines.h, 953
- ewl_engine_theme_freeze
 - ewl_engines.c, 933
- Ewl_Engine_Theme_Hooks
 - ewl_engines.h, 952
- EWL_ENGINE_THEME_MAX
 - ewl_engines.h, 953
- EWL_ENGINE_THEME_OBJECT_ADD
 - ewl_engines.h, 953
- ewl_engine_theme_object_add

- ewl_engines.c, 933
- EWL_ENGINE_THEME_OBJECT_-
CLIP_SET
 - ewl_engines.h, 953
- ewl_engine_theme_object_clip_set
 - ewl_engines.c, 933
- EWL_ENGINE_THEME_OBJECT_-
CLIP_UNSET
 - ewl_engines.h, 953
- ewl_engine_theme_object_clip_unset
 - ewl_engines.c, 934
- EWL_ENGINE_THEME_OBJECT_-
COLOR_CLASS_SET
 - ewl_engines.h, 953
- EWL_ENGINE_THEME_OBJECT_DEL
 - ewl_engines.h, 953
- ewl_engine_theme_object_del
 - ewl_engines.c, 934
- EWL_ENGINE_THEME_OBJECT_-
FILE_SET
 - ewl_engines.h, 953
- ewl_engine_theme_object_file_set
 - ewl_engines.c, 934
- EWL_ENGINE_THEME_OBJECT_HIDE
 - ewl_engines.h, 953
- ewl_engine_theme_object_hide
 - ewl_engines.c, 934
- EWL_ENGINE_THEME_OBJECT_-
LOAD_ERROR
 - ewl_engines.h, 953
- EWL_ENGINE_THEME_OBJECT_-
MAX_SIZE_GET
 - ewl_engines.h, 953
- EWL_ENGINE_THEME_OBJECT_MIN_-
SIZE_GET
 - ewl_engines.h, 953
- EWL_ENGINE_THEME_OBJECT_MOVE
 - ewl_engines.h, 953
- ewl_engine_theme_object_move
 - ewl_engines.c, 934
- EWL_ENGINE_THEME_OBJECT_-
PART_TEXT_SET
 - ewl_engines.h, 953
- EWL_ENGINE_THEME_OBJECT_-
RESIZE
 - ewl_engines.h, 953
- ewl_engine_theme_object_resize
 - ewl_engines.c, 934
- EWL_ENGINE_THEME_OBJECT_SHOW
 - ewl_engines.h, 953
- ewl_engine_theme_object_show
 - ewl_engines.c, 935
- EWL_ENGINE_THEME_OBJECT_-
SIGNAL_SEND
 - ewl_engines.h, 953
- EWL_ENGINE_THEME_THAW
 - ewl_engines.h, 953
- ewl_engine_theme_thaw
 - ewl_engines.c, 935
- EWL_ENGINE_THEME_WIDGET_-
GROUP
 - ewl_engines.h, 953
- ewl_engine_theme_widget_group
 - ewl_engines.c, 935
- EWL_ENGINE_THEME_WIDGET_-
LAYER_UPDATE
 - ewl_engines.h, 953
- EWL_ENGINE_THEME_WIDGET_-
STACK_ADD
 - ewl_engines.h, 953
- EWL_ENGINE_WINDOW_-
BORDERLESS_SET
 - ewl_engines.h, 954
- ewl_engine_window_borderless_set
 - ewl_engines.c, 935
- ewl_engines.h, 959
- EWL_ENGINE_WINDOW_DESKTOP_-
SIZE_GET
 - ewl_engines.h, 954
- EWL_ENGINE_WINDOW_DESTROY
 - ewl_engines.h, 953
- ewl_engine_window_destroy
 - ewl_engines.c, 935
- ewl_engines.h, 959
- EWL_ENGINE_WINDOW_DIALOG_SET
 - ewl_engines.h, 954
- ewl_engine_window_dialog_set
 - ewl_engines.c, 936
- ewl_engines.h, 959
- EWL_ENGINE_WINDOW_DND_-
AWARE_SET
 - ewl_engines.h, 954
- EWL_ENGINE_WINDOW_DND_DRAG_-
DATA_SEND
 - ewl_engines.h, 954
- EWL_ENGINE_WINDOW_DND_DRAG_-
DROP
 - ewl_engines.h, 954
- EWL_ENGINE_WINDOW_DND_DRAG_-
START
 - ewl_engines.h, 954
- EWL_ENGINE_WINDOW_DND_DRAG_-
TYPES_SET
 - ewl_engines.h, 954
- EWL_ENGINE_WINDOW_GEOMETRY_-
GET
 - ewl_engines.h, 954
- ewl_engine_window_geometry_get

- ewl_engines.c, 936
- ewl_engines.h, 960
- EWL_ENGINE_WINDOW_HIDE
 - ewl_engines.h, 953
- ewl_engine_window_hide
 - ewl_engines.c, 936
 - ewl_engines.h, 960
- Ewl_Engine_Window_Hooks
 - ewl_engines.h, 952, 953
- EWL_ENGINE_WINDOW_KEYBOARD_-GRAB
 - ewl_engines.h, 954
- EWL_ENGINE_WINDOW_KEYBOARD_-UNGRAB
 - ewl_engines.h, 954
- EWL_ENGINE_WINDOW_LOWER
 - ewl_engines.h, 954
- ewl_engine_window_lower
 - ewl_engines.c, 936
 - ewl_engines.h, 960
- EWL_ENGINE_WINDOW_MAX
 - ewl_engines.h, 954
- EWL_ENGINE_WINDOW_MIN_MAX_-SIZE_SET
 - ewl_engines.h, 953
- ewl_engine_window_min_max_size_set
 - ewl_engines.c, 937
 - ewl_engines.h, 960
- EWL_ENGINE_WINDOW_MOVE
 - ewl_engines.h, 953
- ewl_engine_window_move
 - ewl_engines.c, 937
 - ewl_engines.h, 961
- EWL_ENGINE_WINDOW_NAME_-CLASS_SET
 - ewl_engines.h, 953
- ewl_engine_window_name_class_set
 - ewl_engines.c, 937
 - ewl_engines.h, 961
- EWL_ENGINE_WINDOW_NEW
 - ewl_engines.h, 953
- ewl_engine_window_new
 - ewl_engines.c, 937
 - ewl_engines.h, 961
- EWL_ENGINE_WINDOW_POINTER_-GRAB
 - ewl_engines.h, 954
- EWL_ENGINE_WINDOW_POINTER_-UNGRAB
 - ewl_engines.h, 954
- EWL_ENGINE_WINDOW_RAISE
 - ewl_engines.h, 954
- ewl_engine_window_raise
 - ewl_engines.c, 938
- ewl_engines.h, 961
- EWL_ENGINE_WINDOW_RESIZE
 - ewl_engines.h, 953
- ewl_engine_window_resize
 - ewl_engines.c, 938
 - ewl_engines.h, 962
- EWL_ENGINE_WINDOW_SELECTION_-TEXT_SET
 - ewl_engines.h, 954
- ewl_engine_window_selection_text_set
 - ewl_engines.c, 938
 - ewl_engines.h, 962
- EWL_ENGINE_WINDOW_SHOW
 - ewl_engines.h, 953
- ewl_engine_window_show
 - ewl_engines.c, 938
 - ewl_engines.h, 962
- EWL_ENGINE_WINDOW_STATES_SET
 - ewl_engines.h, 954
- ewl_engine_window_states_set
 - ewl_engines.c, 939
 - ewl_engines.h, 962
- EWL_ENGINE_WINDOW_TITLE_SET
 - ewl_engines.h, 953
- ewl_engine_window_title_set
 - ewl_engines.c, 939
 - ewl_engines.h, 963
- EWL_ENGINE_WINDOW_TRANSIENT_-FOR
 - ewl_engines.h, 954
- ewl_engine_window_transient_for
 - ewl_engines.c, 939
 - ewl_engines.h, 963
- ewl_engines.c
 - ewl_engine_canvas_damage_add, 928
 - ewl_engine_canvas_freeze, 928
 - ewl_engine_canvas_output_set, 928
 - ewl_engine_canvas_render, 929
 - ewl_engine_canvas_setup, 929
 - ewl_engine_canvas_thaw, 929
 - ewl_engine_desktop_size_get, 929
 - EWL_ENGINE_DIR, 928
 - ewl_engine_embed_dnd_aware_set, 930
 - ewl_engine_embed_dnd_drag_data_-send, 930
 - ewl_engine_embed_dnd_drag_drop, 930
 - ewl_engine_embed_dnd_drag_start, 930
 - ewl_engine_embed_dnd_drag_types_-set, 931
 - Ewl_Engine_Hook_Type, 928
 - EWL_ENGINE_HOOK_TYPE_-CANVAS, 928
 - EWL_ENGINE_HOOK_TYPE_-POINTER, 928

- EWL_ENGINE_HOOK_TYPE_-
THEME, 928
- EWL_ENGINE_HOOK_TYPE_-
WINDOW, 928
- ewl_engine_keyboard_grab, 931
- ewl_engine_keyboard_ungrab, 931
- ewl_engine_names_get, 931
- ewl_engine_new, 932
- ewl_engine_pointer_data_new, 932
- ewl_engine_pointer_free, 932
- ewl_engine_pointer_get, 932
- ewl_engine_pointer_grab, 932
- ewl_engine_pointer_set, 933
- ewl_engine_pointer_ungrab, 933
- ewl_engine_theme_freeze, 933
- ewl_engine_theme_object_add, 933
- ewl_engine_theme_object_clip_set, 933
- ewl_engine_theme_object_clip_unset,
934
- ewl_engine_theme_object_del, 934
- ewl_engine_theme_object_file_set, 934
- ewl_engine_theme_object_hide, 934
- ewl_engine_theme_object_move, 934
- ewl_engine_theme_object_resize, 934
- ewl_engine_theme_object_show, 935
- ewl_engine_theme_thaw, 935
- ewl_engine_theme_widget_group, 935
- ewl_engine_window_borderless_set, 935
- ewl_engine_window_destroy, 935
- ewl_engine_window_dialog_set, 936
- ewl_engine_window_geometry_get, 936
- ewl_engine_window_hide, 936
- ewl_engine_window_lower, 936
- ewl_engine_window_min_max_size_-
set, 937
- ewl_engine_window_move, 937
- ewl_engine_window_name_class_set,
937
- ewl_engine_window_new, 937
- ewl_engine_window_raise, 938
- ewl_engine_window_resize, 938
- ewl_engine_window_selection_text_set,
938
- ewl_engine_window_show, 938
- ewl_engine_window_states_set, 939
- ewl_engine_window_title_set, 939
- ewl_engine_window_transient_for, 939
- ewl_engines_init, 939
- ewl_engines_shutdown, 939
- ewl_engines.h
 - EWL_ENGINE, 947
 - Ewl_Engine, 947
 - EWL_ENGINE_CANVAS_DAMAGE_-
ADD, 952
 - ewl_engine_canvas_damage_add, 954
 - EWL_ENGINE_CANVAS_FREEZE,
952
 - ewl_engine_canvas_freeze, 954
 - Ewl_Engine_Canvas_Hooks, 947, 952
 - EWL_ENGINE_CANVAS_MAX, 952
 - EWL_ENGINE_CANVAS_OUTPUT_-
SET, 952
 - ewl_engine_canvas_output_set, 954
 - EWL_ENGINE_CANVAS_RENDER,
952
 - ewl_engine_canvas_render, 955
 - EWL_ENGINE_CANVAS_SETUP, 952
 - ewl_engine_canvas_setup, 955
 - EWL_ENGINE_CANVAS_THAW, 952
 - ewl_engine_canvas_thaw, 955
 - Ewl_Engine_Cb_Canvas_Damage_Add,
947
 - Ewl_Engine_Cb_Canvas_Freeze, 947
 - Ewl_Engine_Cb_Canvas_Output_Set,
947
 - Ewl_Engine_Cb_Canvas_Render, 947
 - Ewl_Engine_Cb_Canvas_Setup, 947
 - Ewl_Engine_Cb_Canvas_Thaw, 947
 - Ewl_Engine_Cb_Keyboard_Grab, 947
 - Ewl_Engine_Cb_Keyboard_Ungrab, 947
 - Ewl_Engine_Cb_Pointer_Data_New,
947
 - Ewl_Engine_Cb_Pointer_Free, 948
 - Ewl_Engine_Cb_Pointer_Get, 948
 - Ewl_Engine_Cb_Pointer_Grab, 948
 - Ewl_Engine_Cb_Pointer_Set, 948
 - Ewl_Engine_Cb_Pointer_Ungrab, 948
 - Ewl_Engine_Cb_Theme_Clip_Add, 948
 - Ewl_Engine_Cb_Theme_Clip_-
Clipees_Get, 949
 - Ewl_Engine_Cb_Theme_Clip_Color_-
Set, 949
 - Ewl_Engine_Cb_Theme_Clip_Del, 949
 - Ewl_Engine_Cb_Theme_Data_Get, 949
 - Ewl_Engine_Cb_Theme_Freeze, 949
 - Ewl_Engine_Cb_Theme_Object_Add,
949
 - Ewl_Engine_Cb_Theme_Object_-
Clip_Set, 949
 - Ewl_Engine_Cb_Theme_Object_-
Clip_Unset, 949
 - Ewl_Engine_Cb_Theme_Object_-
Color_Class_Set, 949
 - Ewl_Engine_Cb_Theme_Object_Del,
949
 - Ewl_Engine_Cb_Theme_Object_File_-
Set, 949

- Ewl_Engine_Cb_Theme_Object_Hide, 949
- Ewl_Engine_Cb_Theme_Object_-Layer_Update, 949
- Ewl_Engine_Cb_Theme_Object_-Load_Error, 949
- Ewl_Engine_Cb_Theme_Object_-Max_Size_Get, 949
- Ewl_Engine_Cb_Theme_Object_Min_-Size_Get, 949
- Ewl_Engine_Cb_Theme_Object_Move, 949
- Ewl_Engine_Cb_Theme_Object_-Part_Text_Set, 949
- Ewl_Engine_Cb_Theme_Object_-Resize, 949
- Ewl_Engine_Cb_Theme_Object_Show, 949
- Ewl_Engine_Cb_Theme_Object_-Signal_Send, 949
- Ewl_Engine_Cb_Theme_Object_-Stack_Add, 949
- Ewl_Engine_Cb_Theme_Thaw, 949
- Ewl_Engine_Cb_Theme_Widget_-Group, 949
- Ewl_Engine_Cb_Window_Borderless_-Set, 949
- Ewl_Engine_Cb_Window_Desktop_-Size_Get, 949
- Ewl_Engine_Cb_Window_Destroy, 950
- Ewl_Engine_Cb_Window_Dialog_Set, 950
- Ewl_Engine_Cb_Window_Dnd_-Aware_Set, 950
- Ewl_Engine_Cb_Window_Dnd_Drag_-Data_Send, 950
- Ewl_Engine_Cb_Window_Dnd_Drag_-Drop, 950
- Ewl_Engine_Cb_Window_Dnd_Drag_-Start, 950
- Ewl_Engine_Cb_Window_Dnd_Drag_-Types_Set, 950
- Ewl_Engine_Cb_Window_Geometry_-Get, 950
- Ewl_Engine_Cb_Window_Hide, 950
- Ewl_Engine_Cb_Window_Lower, 951
- Ewl_Engine_Cb_Window_Min_Max_-Size_Set, 951
- Ewl_Engine_Cb_Window_Move, 951
- Ewl_Engine_Cb_Window_Name_-Class_Set, 951
- Ewl_Engine_Cb_Window_New, 951
- Ewl_Engine_Cb_Window_Raise, 951
- Ewl_Engine_Cb_Window_Resize, 951
- Ewl_Engine_Cb_Window_Selection_-Text_Set, 951
- Ewl_Engine_Cb_Window_Show, 951
- Ewl_Engine_Cb_Window_States_Set, 951
- Ewl_Engine_Cb_Window_Title_Set, 951
- Ewl_Engine_Cb_Window_Transient_-For, 952
- ewl_engine_desktop_size_get, 955
- ewl_engine_embed_dnd_aware_set, 956
- ewl_engine_embed_dnd_drag_data_-send, 956
- ewl_engine_embed_dnd_drag_drop, 956
- ewl_engine_embed_dnd_drag_start, 956
- ewl_engine_embed_dnd_drag_types_-set, 957
- ewl_engine_engine_shutdown, 957
- ewl_engine_event_handlers_init, 957
- ewl_engine_event_handlers_shutdown, 957
- Ewl_Engine_Info, 952
- ewl_engine_keyboard_grab, 957
- ewl_engine_keyboard_ungrab, 957
- ewl_engine_names_get, 957
- ewl_engine_new, 958
- EWL_ENGINE_POINTER_DATA_-NEW, 952
- ewl_engine_pointer_data_new, 958
- EWL_ENGINE_POINTER_FREE, 952
- ewl_engine_pointer_free, 958
- EWL_ENGINE_POINTER_GET, 952
- ewl_engine_pointer_get, 958
- ewl_engine_pointer_grab, 958
- Ewl_Engine_Pointer_Hooks, 952
- EWL_ENGINE_POINTER_MAX, 952
- EWL_ENGINE_POINTER_SET, 952
- ewl_engine_pointer_set, 959
- ewl_engine_pointer_ungrab, 959
- EWL_ENGINE_THEME_CLIP_ADD, 953
- EWL_ENGINE_THEME_CLIP_-CLIPES_GET, 953
- EWL_ENGINE_THEME_CLIP_-COLOR_SET, 953
- EWL_ENGINE_THEME_CLIP_DEL, 953
- EWL_ENGINE_THEME_DATA_GET, 953
- EWL_ENGINE_THEME_FREEZE, 953
- Ewl_Engine_Theme_Hooks, 952
- EWL_ENGINE_THEME_MAX, 953
- EWL_ENGINE_THEME_OBJECT_-ADD, 953

EWL_ENGINE_THEME_OBJECT_-
 CLIP_SET, 953
 EWL_ENGINE_THEME_OBJECT_-
 CLIP_UNSET, 953
 EWL_ENGINE_THEME_OBJECT_-
 COLOR_CLASS_SET, 953
 EWL_ENGINE_THEME_OBJECT_-
 DEL, 953
 EWL_ENGINE_THEME_OBJECT_-
 FILE_SET, 953
 EWL_ENGINE_THEME_OBJECT_-
 HIDE, 953
 EWL_ENGINE_THEME_OBJECT_-
 LOAD_ERROR, 953
 EWL_ENGINE_THEME_OBJECT_-
 MAX_SIZE_GET, 953
 EWL_ENGINE_THEME_OBJECT_-
 MIN_SIZE_GET, 953
 EWL_ENGINE_THEME_OBJECT_-
 MOVE, 953
 EWL_ENGINE_THEME_OBJECT_-
 PART_TEXT_SET, 953
 EWL_ENGINE_THEME_OBJECT_-
 RESIZE, 953
 EWL_ENGINE_THEME_OBJECT_-
 SHOW, 953
 EWL_ENGINE_THEME_OBJECT_-
 SIGNAL_SEND, 953
 EWL_ENGINE_THEME_THAW, 953
 EWL_ENGINE_THEME_WIDGET_-
 GROUP, 953
 EWL_ENGINE_THEME_WIDGET_-
 LAYER_UPDATE, 953
 EWL_ENGINE_THEME_WIDGET_-
 STACK_ADD, 953
 EWL_ENGINE_WINDOW_-
 BORDERLESS_SET, 954
 ewl_engine_window_borderless_set, 959
 EWL_ENGINE_WINDOW_-
 DESKTOP_SIZE_GET, 954
 EWL_ENGINE_WINDOW_DESTROY,
 953
 ewl_engine_window_destroy, 959
 EWL_ENGINE_WINDOW_DIALOG_-
 SET, 954
 ewl_engine_window_dialog_set, 959
 EWL_ENGINE_WINDOW_DND_-
 AWARE_SET, 954
 EWL_ENGINE_WINDOW_DND_-
 DRAG_DATA_SEND, 954
 EWL_ENGINE_WINDOW_DND_-
 DRAG_DROP, 954
 EWL_ENGINE_WINDOW_DND_-
 DRAG_START, 954
 EWL_ENGINE_WINDOW_DND_-
 DRAG_TYPES_SET, 954
 EWL_ENGINE_WINDOW_-
 GEOMETRY_GET, 954
 ewl_engine_window_geometry_get, 960
 EWL_ENGINE_WINDOW_HIDE, 953
 ewl_engine_window_hide, 960
 Ewl_Engine_Window_Hooks, 952, 953
 EWL_ENGINE_WINDOW_-
 KEYBOARD_GRAB, 954
 EWL_ENGINE_WINDOW_-
 KEYBOARD_UNGRAB, 954
 EWL_ENGINE_WINDOW_LOWER,
 954
 ewl_engine_window_lower, 960
 EWL_ENGINE_WINDOW_MAX, 954
 EWL_ENGINE_WINDOW_MIN_-
 MAX_SIZE_SET, 953
 ewl_engine_window_min_max_size_-
 set, 960
 EWL_ENGINE_WINDOW_MOVE, 953
 ewl_engine_window_move, 961
 EWL_ENGINE_WINDOW_NAME_-
 CLASS_SET, 953
 ewl_engine_window_name_class_set,
 961
 EWL_ENGINE_WINDOW_NEW, 953
 ewl_engine_window_new, 961
 EWL_ENGINE_WINDOW_-
 POINTER_GRAB, 954
 EWL_ENGINE_WINDOW_-
 POINTER_UNGRAB, 954
 EWL_ENGINE_WINDOW_RAISE, 954
 ewl_engine_window_raise, 961
 EWL_ENGINE_WINDOW_RESIZE,
 953
 ewl_engine_window_resize, 962
 EWL_ENGINE_WINDOW_-
 SELECTION_TEXT_SET, 954
 ewl_engine_window_selection_text_set,
 962
 EWL_ENGINE_WINDOW_SHOW, 953
 ewl_engine_window_show, 962
 EWL_ENGINE_WINDOW_STATES_-
 SET, 954
 ewl_engine_window_states_set, 962
 EWL_ENGINE_WINDOW_TITLE_-
 SET, 953
 ewl_engine_window_title_set, 963
 EWL_ENGINE_WINDOW_-
 TRANSIENT_FOR, 954
 ewl_engine_window_transient_for, 963
 ewl_engines_init, 963
 ewl_engines_shutdown, 963

- ewl_engines_init
 - ewl_engines.c, 939
 - ewl_engines.h, 963
- ewl_engines_shutdown
 - ewl_engines.c, 939
 - ewl_engines.h, 963
- EWL_ENTRY
 - Ewl_Entry, 138
- Ewl_Entry, 635
 - cursor, 635
 - editable, 635
 - EWL_ENTRY, 138
 - Ewl_Entry, 139
 - ewl_entry_cb_configure, 139
 - ewl_entry_cb_disable, 139
 - ewl_entry_cb_dnd_data, 139
 - ewl_entry_cb_dnd_position, 139
 - ewl_entry_cb_enable, 139
 - ewl_entry_cb_focus_in, 139
 - ewl_entry_cb_focus_out, 139
 - ewl_entry_cb_key_down, 139
 - ewl_entry_cb_mouse_down, 139
 - ewl_entry_cb_mouse_move, 139
 - ewl_entry_cb_mouse_up, 139
 - EWL_ENTRY_CURSOR, 138
 - Ewl_Entry_Cursor, 139
 - ewl_entry_cursor_init, 139
 - EWL_ENTRY_CURSOR_IS, 138
 - ewl_entry_cursor_move_down, 140
 - ewl_entry_cursor_move_left, 140
 - ewl_entry_cursor_move_right, 140
 - ewl_entry_cursor_move_up, 140
 - ewl_entry_cursor_new, 140
 - ewl_entry_cursor_position_get, 141
 - ewl_entry_cursor_position_set, 141
 - EWL_ENTRY_CURSOR_TYPE, 138
 - ewl_entry_delete_left, 141
 - ewl_entry_delete_right, 141
 - ewl_entry_editable_get, 142
 - ewl_entry_editable_set, 142
 - ewl_entry_init, 142
 - EWL_ENTRY_IS, 138
 - ewl_entry_multiline_get, 142
 - ewl_entry_multiline_set, 143
 - ewl_entry_new, 143
 - ewl_entry_selection_clear, 143
 - EWL_ENTRY_TYPE, 139
 - in_select_mode, 635
 - multiline, 636
 - text, 636
- Ewl_Entry: A text entry widget, 136
- ewl_entry_cb_configure
 - Ewl_Entry, 139
 - lib/ewl_entry.c, 966
- ewl_entry_cb_disable
 - Ewl_Entry, 139
 - lib/ewl_entry.c, 966
- ewl_entry_cb_dnd_data
 - Ewl_Entry, 139
 - lib/ewl_entry.c, 966
- ewl_entry_cb_dnd_position
 - Ewl_Entry, 139
 - lib/ewl_entry.c, 966
- ewl_entry_cb_enable
 - Ewl_Entry, 139
 - lib/ewl_entry.c, 966
- ewl_entry_cb_focus_in
 - Ewl_Entry, 139
 - lib/ewl_entry.c, 966
- ewl_entry_cb_focus_out
 - Ewl_Entry, 139
 - lib/ewl_entry.c, 966
- ewl_entry_cb_key_down
 - Ewl_Entry, 139
 - lib/ewl_entry.c, 966
- ewl_entry_cb_mouse_down
 - Ewl_Entry, 139
 - lib/ewl_entry.c, 966
- ewl_entry_cb_mouse_move
 - Ewl_Entry, 139
 - lib/ewl_entry.c, 966
- ewl_entry_cb_mouse_up
 - Ewl_Entry, 139
 - lib/ewl_entry.c, 966
- EWL_ENTRY_CURSOR
 - Ewl_Entry, 138
- Ewl_Entry_Cursor, 637
 - Ewl_Entry, 139
 - parent, 637
 - widget, 637
- ewl_entry_cursor_init
 - Ewl_Entry, 139
- EWL_ENTRY_CURSOR_IS
 - Ewl_Entry, 138
- ewl_entry_cursor_move_down
 - Ewl_Entry, 140
- ewl_entry_cursor_move_left
 - Ewl_Entry, 140
- ewl_entry_cursor_move_right
 - Ewl_Entry, 140
- ewl_entry_cursor_move_up
 - Ewl_Entry, 140
- ewl_entry_cursor_new
 - Ewl_Entry, 140
- ewl_entry_cursor_position_get
 - Ewl_Entry, 141
- ewl_entry_cursor_position_set
 - Ewl_Entry, 141

- EWL_ENTRY_CURSOR_TYPE
 - Ewl_Entry, 138
- ewl_entry_delete_left
 - Ewl_Entry, 141
- ewl_entry_delete_right
 - Ewl_Entry, 141
- ewl_entry_editable_get
 - Ewl_Entry, 142
- ewl_entry_editable_set
 - Ewl_Entry, 142
- ewl_entry_init
 - Ewl_Entry, 142
- EWL_ENTRY_IS
 - Ewl_Entry, 138
- ewl_entry_multiline_get
 - Ewl_Entry, 142
- ewl_entry_multiline_set
 - Ewl_Entry, 143
- ewl_entry_new
 - Ewl_Entry, 143
- ewl_entry_selection_clear
 - Ewl_Entry, 143
- EWL_ENTRY_TYPE
 - Ewl_Entry, 139
- Ewl_Enums
 - Ewl_Attach_Data_Type, 151, 154
 - EWL_ATTACH_DATA_TYPE_-OTHER, 154
 - EWL_ATTACH_DATA_TYPE_TEXT, 154
 - EWL_ATTACH_DATA_TYPE_-WIDGET, 154
 - Ewl_Attach_Type, 151, 154
 - EWL_ATTACH_TYPE_COLOR, 154
 - EWL_ATTACH_TYPE_DND_DATA, 154
 - EWL_ATTACH_TYPE_MOUSE_-ARGB_CURSOR, 154
 - EWL_ATTACH_TYPE_MOUSE_-CURSOR, 154
 - EWL_ATTACH_TYPE_NAME, 154
 - EWL_ATTACH_TYPE_TOOLTIP, 154
 - EWL_ATTACH_TYPE_WIDGET_-ASSOCIATION, 154
 - EWL_CALLBACK_CLICKED, 155
 - EWL_CALLBACK_CONFIGURE, 154
 - EWL_CALLBACK_DESTROY, 154
 - EWL_CALLBACK_FOCUS_IN, 155
 - EWL_CALLBACK_FOCUS_OUT, 155
 - EWL_CALLBACK_HIDE, 154
 - EWL_CALLBACK_KEY_DOWN, 155
 - EWL_CALLBACK_KEY_UP, 155
 - EWL_CALLBACK_MAX, 155
 - EWL_CALLBACK_MOUSE_DOWN, 155
 - EWL_CALLBACK_MOUSE_IN, 155
 - EWL_CALLBACK_MOUSE_MOVE, 155
 - EWL_CALLBACK_MOUSE_OUT, 155
 - EWL_CALLBACK_MOUSE_UP, 155
 - EWL_CALLBACK_MOUSE_WHEEL, 155
 - EWL_CALLBACK_NOTIFY_-INTERCEPT, 155
 - EWL_CALLBACK_NOTIFY_NONE, 155
 - EWL_CALLBACK_NOTIFY_NOTIFY, 155
 - EWL_CALLBACK_OBSCURE, 154
 - EWL_CALLBACK_REALIZE, 154
 - EWL_CALLBACK_REPARENT, 155
 - EWL_CALLBACK_REVEAL, 154
 - EWL_CALLBACK_SHOW, 154
 - EWL_CALLBACK_STATE_-CHANGED, 155
 - Ewl_Callback_Type, 151, 154
 - EWL_CALLBACK_TYPE_DIRECT, 155
 - EWL_CALLBACK_UNREALIZE, 154
 - EWL_CALLBACK_VALUE_-CHANGED, 155
 - EWL_CALLBACK_WIDGET_-DISABLE, 155
 - EWL_CALLBACK_WIDGET_-ENABLE, 155
 - Ewl_Color_Mode, 151, 155
 - EWL_COLOR_MODE_HSV_HUE, 155
 - EWL_COLOR_MODE_HSV_-SATURATION, 155
 - EWL_COLOR_MODE_HSV_VALUE, 155
 - EWL_COLOR_MODE_RGB_BLUE, 155
 - EWL_COLOR_MODE_RGB_GREEN, 155
 - EWL_COLOR_MODE_RGB_RED, 155
 - Ewl_Event_Notify, 151, 155
 - EWL_FILELIST_EVENT_DIR_-CHANGE, 156
 - EWL_FILELIST_EVENT_FILE_-SELECTED, 156
 - EWL_FILELIST_EVENT_-SELECTION_CHANGE, 156
 - Ewl_Filelist_Event_Type, 151, 155
 - EWL_FLAG_ALIGN_BOTTOM, 156
 - EWL_FLAG_ALIGN_CENTER, 156
 - EWL_FLAG_ALIGN_LEFT, 156

- EWL_FLAG_ALIGN_RIGHT, 156
- EWL_FLAG_ALIGN_TOP, 156
- EWL_FLAG_FILL_ALL, 156
- EWL_FLAG_FILL_FILL, 156
- EWL_FLAG_FILL_HFILL, 156
- EWL_FLAG_FILL_HSHRINK, 156
- EWL_FLAG_FILL_NONE, 156
- EWL_FLAG_FILL_NORMAL, 149
- EWL_FLAG_FILL_SHRINK, 156
- EWL_FLAG_FILL_VFILL, 156
- EWL_FLAG_FILL_VSHRINK, 156
- EWL_FLAG_PROPERTY_BLOCK -
TAB_FOCUS, 156
- EWL_FLAG_PROPERTY_DND -
SOURCE, 156
- EWL_FLAG_PROPERTY_DND -
TARGET, 156
- EWL_FLAG_PROPERTY -
FOCUSABLE, 156
- EWL_FLAG_PROPERTY_IN_TAB -
LIST, 156
- EWL_FLAG_PROPERTY -
INTERNAL, 156
- EWL_FLAG_PROPERTY -
RECURSIVE, 156
- EWL_FLAG_PROPERTY -
TOPLEVEL, 156
- EWL_FLAG_QUEUED_CPROCESS,
157
- EWL_FLAG_QUEUED -
CSCHEDULED, 156
- EWL_FLAG_QUEUED_DPROCESS,
157
- EWL_FLAG_QUEUED -
DSCHEDULED, 156
- EWL_FLAG_QUEUED_RPROCESS,
157
- EWL_FLAG_QUEUED -
RSCHEDULED, 156
- EWL_FLAG_STATE_DISABLED, 157
- EWL_FLAG_STATE_DND, 157
- EWL_FLAG_STATE_DND_WAIT, 157
- EWL_FLAG_STATE_FOCUSED, 157
- EWL_FLAG_STATE_MOUSE_IN, 157
- EWL_FLAG_STATE_NORMAL, 157
- EWL_FLAG_STATE_PRESSED, 157
- EWL_FLAG_VISIBLE_HIDDEN, 156
- EWL_FLAG_VISIBLE_NOCLIP, 156
- EWL_FLAG_VISIBLE_OBSCURED,
156
- EWL_FLAG_VISIBLE_REALIZED, 156
- EWL_FLAG_VISIBLE_SHOWN, 156
- Ewl_Flags, 156
- EWL_FLAGS_ALIGN_MASK, 149
- EWL_FLAGS_FILL_MASK, 150
- EWL_FLAGS_PROPERTY_MASK, 150
- EWL_FLAGS_QUEUED_MASK, 150
- EWL_FLAGS_STATE_MASK, 150
- EWL_FLAGS_VISIBLE_MASK, 150
- EWL_FREEBOX_LAYOUT_AUTO,
157
- EWL_FREEBOX_LAYOUT -
COMPARATOR, 157
- EWL_FREEBOX_LAYOUT -
MANUAL, 157
- Ewl_Freebox_Layout_Type, 151, 157
- EWL_GRID_RESIZE_FIXED, 157
- EWL_GRID_RESIZE_NONE, 157
- EWL_GRID_RESIZE_NORMAL, 157
- EWL_GRID_RESIZE_RELATIVE, 157
- Ewl_Grid_Resize_Type, 151, 157
- Ewl_Histogram_Channel, 151, 157
- EWL_HISTOGRAM_CHANNEL_B,
157
- EWL_HISTOGRAM_CHANNEL_G,
157
- EWL_HISTOGRAM_CHANNEL -
MAX, 157
- EWL_HISTOGRAM_CHANNEL_R,
157
- EWL_HISTOGRAM_CHANNEL_Y,
157
- Ewl_Icon_Part, 152, 157
- EWL_ICON_PART_IMAGE, 158
- EWL_ICON_PART_LABEL, 158
- EWL_ICON_PART_NONE, 158
- Ewl_Icon_Type, 152, 158
- EWL_ICON_TYPE_LONG, 158
- EWL_ICON_TYPE_SHORT, 158
- Ewl_Image_Type, 152, 158
- EWL_IMAGE_TYPE_EDJE, 158
- EWL_IMAGE_TYPE_NORMAL, 158
- EWL_KEY_MODIFIER_ALT, 158
- EWL_KEY_MODIFIER_CTRL, 158
- EWL_KEY_MODIFIER_MOD, 158
- EWL_KEY_MODIFIER_SHIFT, 158
- EWL_KEY_MODIFIER_WIN, 158
- Ewl_Key_Modifiers, 152, 158
- EWL_MEDIA_MODULE -
GSTREAMER, 158
- Ewl_Media_Module_Type, 152, 158
- EWL_MEDIA_MODULE_XINE, 158
- EWL_MOUSE_CURSOR_ARROW, 159
- EWL_MOUSE_CURSOR_BASED -
ARROW_DOWN, 159
- EWL_MOUSE_CURSOR_BOAT, 159
- EWL_MOUSE_CURSOR_BOTTOM -
LEFT_CORNER, 159

- EWL_MOUSE_CURSOR_BOTTOM_-
RIGHT_CORNER, 159
- EWL_MOUSE_CURSOR_BOTTOM_-
SIDE, 159
- EWL_MOUSE_CURSOR_BOTTOM_-
TEE, 159
- EWL_MOUSE_CURSOR_BOX_-
SPIRAL, 159
- EWL_MOUSE_CURSOR_CENTER_-
PTR, 159
- EWL_MOUSE_CURSOR_CIRCLE, 159
- EWL_MOUSE_CURSOR_CLOCK, 159
- EWL_MOUSE_CURSOR_COFFEE_-
MUG, 159
- EWL_MOUSE_CURSOR_CROSS, 159
- EWL_MOUSE_CURSOR_CROSS_-
REVERSE, 159
- EWL_MOUSE_CURSOR_-
CROSSHAIR, 159
- EWL_MOUSE_CURSOR_-
DIAMOND_CROSS, 159
- EWL_MOUSE_CURSOR_DOT, 159
- EWL_MOUSE_CURSOR_DOT_-
BOX_MASK, 159
- EWL_MOUSE_CURSOR_DOUBLE_-
ARROW, 159
- EWL_MOUSE_CURSOR_DRAFT_-
LARGE, 159
- EWL_MOUSE_CURSOR_DRAFT_-
SMALL, 159
- EWL_MOUSE_CURSOR_DRAPED_-
BOX, 159
- EWL_MOUSE_CURSOR_-
EXCHANGE, 159
- EWL_MOUSE_CURSOR_FLEUR, 159
- EWL_MOUSE_CURSOR_GOBLER,
159
- EWL_MOUSE_CURSOR_GUMBY, 159
- EWL_MOUSE_CURSOR_HAND1, 159
- EWL_MOUSE_CURSOR_HAND2, 159
- EWL_MOUSE_CURSOR_HEART, 159
- EWL_MOUSE_CURSOR_ICON, 159
- EWL_MOUSE_CURSOR_IRON_-
CROSS, 159
- EWL_MOUSE_CURSOR_LEFT_PTR,
159
- EWL_MOUSE_CURSOR_LEFT_SIDE,
159
- EWL_MOUSE_CURSOR_LEFT_TEE,
159
- EWL_MOUSE_CURSOR_-
LEFTBUTTON, 159
- EWL_MOUSE_CURSOR_LL_ANGLE,
159
- EWL_MOUSE_CURSOR_LR_ANGLE,
160
- EWL_MOUSE_CURSOR_MAN, 160
- EWL_MOUSE_CURSOR_MAX, 160
- EWL_MOUSE_CURSOR_-
MIDDLEBUTTON, 160
- EWL_MOUSE_CURSOR_MOUSE, 160
- EWL_MOUSE_CURSOR_PENCIL, 160
- EWL_MOUSE_CURSOR_PIRATE, 160
- EWL_MOUSE_CURSOR_PLUS, 160
- EWL_MOUSE_CURSOR_-
QUESTION_ARROW, 160
- EWL_MOUSE_CURSOR_RIGHT_-
PTR, 160
- EWL_MOUSE_CURSOR_RIGHT_-
SIDE, 160
- EWL_MOUSE_CURSOR_RIGHT_-
TEE, 160
- EWL_MOUSE_CURSOR_-
RIGHTBUTTON, 160
- EWL_MOUSE_CURSOR_RTL_LOGO,
160
- EWL_MOUSE_CURSOR_SAILBOAT,
160
- EWL_MOUSE_CURSOR_SB_-
DOWN_ARROW, 160
- EWL_MOUSE_CURSOR_SB_H_-
DOUBLE_ARROW, 160
- EWL_MOUSE_CURSOR_SB_LEFT_-
ARROW, 160
- EWL_MOUSE_CURSOR_SB_-
RIGHT_ARROW, 160
- EWL_MOUSE_CURSOR_SB_UP_-
ARROW, 160
- EWL_MOUSE_CURSOR_SB_V_-
DOUBLE_ARROW, 160
- EWL_MOUSE_CURSOR_SHUTTLE,
160
- EWL_MOUSE_CURSOR_SIZING, 160
- EWL_MOUSE_CURSOR_SPIDER, 160
- EWL_MOUSE_CURSOR_SPRAYCAN,
160
- EWL_MOUSE_CURSOR_STAR, 160
- EWL_MOUSE_CURSOR_TARGET,
160
- EWL_MOUSE_CURSOR_TCROSS, 160
- EWL_MOUSE_CURSOR_TOP_-
LEFT_ARROW, 160
- EWL_MOUSE_CURSOR_TOP_-
LEFT_CORNER, 160
- EWL_MOUSE_CURSOR_TOP_-
RIGHT_CORNER, 160
- EWL_MOUSE_CURSOR_TOP_SIDE,
160

- EWL_MOUSE_CURSOR_TOP_TEE, 160
- EWL_MOUSE_CURSOR_TREK, 160
- Ewl_Mouse_Cursor_Type, 152, 158
- EWL_MOUSE_CURSOR_UL_ANGLE, 160
- EWL_MOUSE_CURSOR_UMBRELLA, 160
- EWL_MOUSE_CURSOR_UP, 159
- EWL_MOUSE_CURSOR_UR_-ANGLE, 160
- EWL_MOUSE_CURSOR_WATCH, 160
- EWL_MOUSE_CURSOR_X, 159
- EWL_MOUSE_CURSOR_XTERM, 160
- EWL_NOTEBOOK_FLAG_TABS_-HIDDEN, 161
- Ewl_Notebook_Flags, 152, 160
- Ewl_Orientation, 152, 161
- EWL_ORIENTATION_HORIZONTAL, 161
- EWL_ORIENTATION_VERTICAL, 161
- Ewl_Popup_Type, 152, 161
- EWL_POPUP_TYPE_MENU_-HORIZONTAL, 161
- EWL_POPUP_TYPE_MENU_-VERTICAL, 161
- EWL_POPUP_TYPE_MOUSE, 161
- EWL_POPUP_TYPE_NONE, 161
- Ewl_Position, 152, 161
- EWL_POSITION_BOTTOM, 161
- EWL_POSITION_LEFT, 161
- EWL_POSITION_MASK, 151
- EWL_POSITION_RIGHT, 161
- EWL_POSITION_TOP, 161
- Ewl_Rotate, 152, 161
- EWL_ROTATE_180, 161
- EWL_ROTATE_CC_270, 161
- EWL_ROTATE_CC_90, 161
- EWL_ROTATE_CW_270, 161
- EWL_ROTATE_CW_90, 161
- EWL_SCROLLPANE_FLAG_-ALWAYS_HIDDEN, 162
- EWL_SCROLLPANE_FLAG_AUTO_-VISIBLE, 162
- EWL_SCROLLPANE_FLAG_NONE, 162
- Ewl_Scrollpane_Flags, 152, 161
- Ewl_Selection_Mode, 153, 162
- EWL_SELECTION_MODE_MULTI, 162
- EWL_SELECTION_MODE_NONE, 162
- EWL_SELECTION_MODE_SINGLE, 162
- Ewl_Selection_Type, 153, 162
- EWL_SELECTION_TYPE_INDEX, 162
- EWL_SELECTION_TYPE_RANGE, 162
- Ewl_Sort_Direction, 153, 162
- EWL_SORT_DIRECTION_-ASCENDING, 162
- EWL_SORT_DIRECTION_-DESCENDING, 162
- EWL_SORT_DIRECTION_MAX, 162
- EWL_SORT_DIRECTION_NONE, 162
- Ewl_Spectrum_Type, 153, 162
- EWL_SPECTRUM_TYPE_SQUARE, 162
- EWL_SPECTRUM_TYPE_-VERTICAL, 162
- EWL_STATE_PERSISTENT, 163
- EWL_STATE_TRANSIENT, 163
- Ewl_State_Type, 153, 162
- EWL_STOCK_APPLY, 163
- EWL_STOCK_ARROW_DOWN, 163
- EWL_STOCK_ARROW_LEFT, 163
- EWL_STOCK_ARROW_RIGHT, 163
- EWL_STOCK_ARROW_UP, 163
- EWL_STOCK_CANCEL, 163
- EWL_STOCK_FASTFORWARD, 163
- EWL_STOCK_HOME, 163
- EWL_STOCK_NONE, 163
- EWL_STOCK_OK, 163
- EWL_STOCK_OPEN, 163
- EWL_STOCK_PAUSE, 163
- EWL_STOCK_PLAY, 163
- EWL_STOCK_QUIT, 163
- EWL_STOCK_REWIND, 163
- EWL_STOCK_SAVE, 163
- EWL_STOCK_STOP, 163
- Ewl_Stock_Type, 153, 163
- Ewl_Text_Context_Mask, 153, 163
- EWL_TEXT_CONTEXT_MASK_-ALIGN, 163
- EWL_TEXT_CONTEXT_MASK_-BG_COLOR, 163
- EWL_TEXT_CONTEXT_MASK_-COLOR, 163
- EWL_TEXT_CONTEXT_MASK_-DOUBLE_UNDERLINE_COLOR, 164
- EWL_TEXT_CONTEXT_MASK_-FONT, 163
- EWL_TEXT_CONTEXT_MASK_-GLOW_COLOR, 164
- EWL_TEXT_CONTEXT_MASK_-NONE, 163
- EWL_TEXT_CONTEXT_MASK_-OUTLINE_COLOR, 164

- EWL_TEXT_CONTEXT_MASK_-
SHADOW_COLOR, 164
- EWL_TEXT_CONTEXT_MASK_-
SIZE, 163
- EWL_TEXT_CONTEXT_MASK_-
STRIKETHROUGH_COLOR, 164
- EWL_TEXT_CONTEXT_MASK_-
STYLES, 163
- EWL_TEXT_CONTEXT_MASK_-
UNDERLINE_COLOR, 164
- EWL_TEXT_CONTEXT_MASK_-
WRAP, 163
- Ewl_Text_Style, 153, 164
- EWL_TEXT_STYLE_DOUBLE_-
UNDERLINE, 164
- EWL_TEXT_STYLE_FAR_SHADOW,
164
- EWL_TEXT_STYLE_GLOW, 164
- EWL_TEXT_STYLE_NONE, 164
- EWL_TEXT_STYLE_OUTLINE, 164
- EWL_TEXT_STYLE_SHADOW, 164
- EWL_TEXT_STYLE_SOFT_-
SHADOW, 164
- EWL_TEXT_STYLE_-
STRIKETHROUGH, 164
- EWL_TEXT_STYLE_UNDERLINE,
164
- Ewl_Text_Trigger_Type, 153, 164
- EWL_TEXT_TRIGGER_TYPE_-
NONE, 164
- EWL_TEXT_TRIGGER_TYPE_-
SELECTION, 164
- EWL_TEXT_TRIGGER_TYPE_-
TRIGGER, 164
- Ewl_Text_Wrap, 153, 164
- EWL_TEXT_WRAP_CHAR, 164
- EWL_TEXT_WRAP_NONE, 164
- EWL_TEXT_WRAP_WORD, 164
- EWL_TREE_NODE_COLLAPSED, 165
- EWL_TREE_NODE_EXPANDED, 165
- Ewl_Tree_Node_Flags, 153, 164
- EWL_TREE_NODE_NOEXPAND, 165
- Ewl_Tree_Selection_Type, 153, 165
- EWL_TREE_SELECTION_TYPE_-
CELL, 165
- EWL_TREE_SELECTION_TYPE_-
ROW, 165
- EWL_WINDOW_BORDERLESS, 165
- EWL_WINDOW_DEMANDS_-
ATTENTION, 165
- EWL_WINDOW_DIALOG, 165
- Ewl_Window_Flags, 154, 165
- EWL_WINDOW_FULLSCREEN, 165
- EWL_WINDOW_GRAB_KEYBOARD,
165
- EWL_WINDOW_GRAB_POINTER,
165
- EWL_WINDOW_MODAL, 165
- EWL_WINDOW_OVERRIDE, 165
- EWL_WINDOW_SKIP_PAGER, 165
- EWL_WINDOW_SKIP_TASKBAR, 165
- EWL_WINDOW_TRANSIENT, 165
- EWL_WINDOW_TRANSIENT_-
FOREIGN, 165
- EWL_WINDOW_USER_CONFIGURE,
165
- Ewl_Enums: Various Flags and Enumerations
used in EWL, 144
- ewl_ev_modifiers_get
Ewl_Events, 169
- ewl_ev_modifiers_set
Ewl_Events, 169
- Ewl_Event_Action_Response, 639
Ewl_Events, 167
response, 639
- Ewl_Event_Dnd_Data_Received, 640
data, 640
Ewl_Events, 167
format, 640
len, 640
type, 640
- Ewl_Event_Dnd_Data_Request, 641
Ewl_Events, 168
handle, 641
type, 641
- Ewl_Event_Dnd_Drop, 642
data, 642
Ewl_Events, 168
x, 642
y, 642
- Ewl_Event_Dnd_Position, 643
Ewl_Events, 168
x, 643
y, 643
- Ewl_Event_Key, 644
Ewl_Events, 168
keyname, 644
modifiers, 644
- Ewl_Event_Key_Down, 645
base, 645
Ewl_Events, 168
- Ewl_Event_Key_Up, 646
base, 646
Ewl_Events, 168
- Ewl_Event_Mouse, 647
Ewl_Events, 168
modifiers, 647

- x, 647
- y, 647
- Ewl_Event_Mouse_Down, 648
 - base, 648
 - button, 648
 - clicks, 648
 - Ewl_Events, 168
- Ewl_Event_Mouse_In, 649
 - base, 649
 - Ewl_Events, 168
- Ewl_Event_Mouse_Move, 650
 - base, 650
 - Ewl_Events, 168
- Ewl_Event_Mouse_Out, 651
 - base, 651
 - Ewl_Events, 168
- Ewl_Event_Mouse_Up, 652
 - base, 652
 - button, 652
 - Ewl_Events, 169
- Ewl_Event_Mouse_Wheel, 653
 - base, 653
 - dir, 653
 - Ewl_Events, 169
 - z, 653
- Ewl_Event_Notify
 - Ewl_Enums, 151, 155
- Ewl_Event_Window_Expose, 654
 - Ewl_Events, 169
 - h, 654
 - w, 654
 - x, 654
 - y, 654
- Ewl_Events
 - Ewl_Dnd_Types, 167
 - ewl_ev_modifiers_get, 169
 - ewl_ev_modifiers_set, 169
 - Ewl_Event_Action_Response, 167
 - Ewl_Event_Dnd_Data_Received, 167
 - Ewl_Event_Dnd_Data_Request, 168
 - Ewl_Event_Dnd_Drop, 168
 - Ewl_Event_Dnd_Position, 168
 - Ewl_Event_Key, 168
 - Ewl_Event_Key_Down, 168
 - Ewl_Event_Key_Up, 168
 - Ewl_Event_Mouse, 168
 - Ewl_Event_Mouse_Down, 168
 - Ewl_Event_Mouse_In, 168
 - Ewl_Event_Mouse_Move, 168
 - Ewl_Event_Mouse_Out, 168
 - Ewl_Event_Mouse_Up, 169
 - Ewl_Event_Mouse_Wheel, 169
 - Ewl_Event_Window_Expose, 169
- ewl_events.c
 - ewl_embed_list, 977
- Ewl_Events: Lower Level Event Handlers, 166
- EWL_EXPANSION
 - Ewl_Expansion, 171
- Ewl_Expansion, 655
 - EWL_EXPANSION, 171
 - Ewl_Expansion, 171
 - ewl_expansion_cb_reveal, 171
 - ewl_expansion_cb_update_expandable, 171
 - ewl_expansion_expandable_set, 171
 - ewl_expansion_expanded_set, 171
 - ewl_expansion_init, 172
 - EWL_EXPANSION_IS, 171
 - ewl_expansion_is_expandable, 172
 - ewl_expansion_is_expanded, 172
 - ewl_expansion_new, 172
 - EWL_EXPANSION_TYPE, 171
 - expandable, 655
 - w, 655
- ewl_expansion.c
 - ewl_expansion_cb_reveal, 981
 - ewl_expansion_cb_update_expandable, 981
- Ewl_Expansion: The Expansion node for Tree2, 170
- ewl_expansion_cb_reveal
 - Ewl_Expansion, 171
 - ewl_expansion.c, 981
- ewl_expansion_cb_update_expandable
 - Ewl_Expansion, 171
 - ewl_expansion.c, 981
- ewl_expansion_expandable_set
 - Ewl_Expansion, 171
- ewl_expansion_expanded_set
 - Ewl_Expansion, 171
- ewl_expansion_init
 - Ewl_Expansion, 172
- EWL_EXPANSION_IS
 - Ewl_Expansion, 171
- ewl_expansion_is_expandable
 - Ewl_Expansion, 172
- ewl_expansion_is_expanded
 - Ewl_Expansion, 172
- ewl_expansion_new
 - Ewl_Expansion, 172
- EWL_EXPANSION_TYPE
 - Ewl_Expansion, 171
- EWL_FILEDIALOG
 - Ewl_Filedialog, 174
- Ewl_Filedialog, 656
 - dialog, 657
 - EWL_FILEDIALOG, 174
 - Ewl_Filedialog, 175

- ewl_filedialog_cb_delete_window, 175
- ewl_filedialog_directory_get, 175
- ewl_filedialog_directory_set, 175
- ewl_filedialog_filter_add, 175
- ewl_filedialog_init, 175
- EWL_FILEDIALOG_IS, 174
- ewl_filedialog_list_view_get, 176
- ewl_filedialog_list_view_set, 176
- ewl_filedialog_multiselect_get, 176
- ewl_filedialog_multiselect_new, 176
- ewl_filedialog_multiselect_set, 177
- ewl_filedialog_new, 177
- ewl_filedialog_selected_file_get, 177
- ewl_filedialog_selected_file_set, 177
- ewl_filedialog_selected_files_get, 177
- ewl_filedialog_selected_files_set, 178
- ewl_filedialog_show_dot_files_get, 178
- ewl_filedialog_show_dot_files_set, 178
- ewl_filedialog_show_favorites_get, 178
- ewl_filedialog_show_favorites_set, 179
- EWL_FILEDIALOG_TYPE, 174
- fp, 657
- menu, 657
- Ewl_Filedialog: A Dialog For Picking Files, 173
- ewl_filedialog_cb_delete_window
 - Ewl_Filedialog, 175
 - lib/ewl_filedialog.c, 985
- ewl_filedialog_directory_get
 - Ewl_Filedialog, 175
- ewl_filedialog_directory_set
 - Ewl_Filedialog, 175
- ewl_filedialog_filter_add
 - Ewl_Filedialog, 175
- ewl_filedialog_init
 - Ewl_Filedialog, 175
- EWL_FILEDIALOG_IS
 - Ewl_Filedialog, 174
- ewl_filedialog_list_view_get
 - Ewl_Filedialog, 176
- ewl_filedialog_list_view_set
 - Ewl_Filedialog, 176
- ewl_filedialog_multiselect_get
 - Ewl_Filedialog, 176
- ewl_filedialog_multiselect_new
 - Ewl_Filedialog, 176
- ewl_filedialog_multiselect_set
 - Ewl_Filedialog, 177
- ewl_filedialog_new
 - Ewl_Filedialog, 177
- ewl_filedialog_selected_file_get
 - Ewl_Filedialog, 177
- ewl_filedialog_selected_file_set
 - Ewl_Filedialog, 177
- ewl_filedialog_selected_files_get
 - Ewl_Filedialog, 177
- ewl_filedialog_selected_files_set
 - Ewl_Filedialog, 178
- ewl_filedialog_show_dot_files_get
 - Ewl_Filedialog, 178
- ewl_filedialog_show_dot_files_set
 - Ewl_Filedialog, 178
- ewl_filedialog_show_favorites_get
 - Ewl_Filedialog, 178
- ewl_filedialog_show_favorites_set
 - Ewl_Filedialog, 179
- EWL_FILEDIALOG_TYPE
 - Ewl_Filedialog, 174
- EWL_FILELIST
 - Ewl_Filelist, 182
- Ewl_Filelist, 658
 - base, 659
 - box, 659
 - dir_change, 659
 - directory, 659
 - EWL_FILELIST, 182
 - Ewl_Filelist, 182
 - ewl_filelist_cb_destroy, 183
 - ewl_filelist_container_shift_handle, 183
 - ewl_filelist_directory_get, 183
 - ewl_filelist_directory_read, 183
 - ewl_filelist_directory_set, 183
 - ewl_filelist_expand_path, 183
 - ewl_filelist_filter_get, 183
 - ewl_filelist_filter_set, 184
 - ewl_filelist_groupname_get, 184
 - ewl_filelist_handle_click, 184
 - ewl_filelist_hscroll_flag_get, 184
 - ewl_filelist_hscroll_flag_set, 184
 - ewl_filelist_init, 185
 - EWL_FILELIST_IS, 182
 - ewl_filelist_modtime_get, 185
 - ewl_filelist_multiselect_get, 185
 - ewl_filelist_multiselect_set, 185
 - ewl_filelist_perms_get, 186
 - ewl_filelist_selected_file_get, 186
 - ewl_filelist_selected_file_preview_get, 186
 - ewl_filelist_selected_file_set, 186
 - ewl_filelist_selected_files_change_notify, 187
 - ewl_filelist_selected_files_get, 187
 - ewl_filelist_selected_files_set, 187
 - ewl_filelist_selected_signal_all, 187
 - ewl_filelist_show_dot_files_get, 188
 - ewl_filelist_show_dot_files_set, 188
 - ewl_filelist_size_get, 188
 - ewl_filelist_stock_icon_get, 188
 - EWL_FILELIST_TYPE, 182

- ewl_filelist_username_get, 189
- ewl_filelist_vscroll_flag_get, 189
- ewl_filelist_vscroll_flag_set, 189
- file_name_get, 659
- filter, 659
- filter_change, 659
- h, 659
- last, 659
- multiselect, 659
- multiselect_change, 659
- scroll_flags, 660
- select, 660
- selected, 660
- selected_file_add, 660
- selected_unselect, 660
- shift_handle, 660
- show_dot_change, 660
- show_dot_files, 660
- v, 660
- ewl_filelist.c
 - ewl_filelist_cb_destroy, 991
- Ewl_Filelist: The base widget for the filelists, 180
- ewl_filelist_cb_destroy
 - Ewl_Filelist, 183
 - ewl_filelist.c, 991
- EWL_FILELIST_COLUMN
 - Ewl_Filelist_Column, 191
- Ewl_Filelist_Column, 661
 - dirs, 661
 - EWL_FILELIST_COLUMN, 191
 - Ewl_Filelist_Column, 191
 - ewl_filelist_column_dir_change, 191
 - ewl_filelist_column_filename_get, 191
 - ewl_filelist_column_init, 191
 - EWL_FILELIST_COLUMN_IS, 191
 - ewl_filelist_column_new, 192
 - ewl_filelist_column_selected_file_add, 192
 - ewl_filelist_column_selected_unselect, 192
 - ewl_filelist_column_shift_handle, 192
 - EWL_FILELIST_COLUMN_TYPE, 191
 - ewl_filelist_column_view_get, 192
 - hbox, 661
 - list, 661
 - preview, 662
- ewl_filelist_column.c
 - Ewl_Filelist_Column_Data, 996
- Ewl_Filelist_Column: An expanding column view, 190
- Ewl_Filelist_Column_Data, 663
 - dir, 663
 - ewl_filelist_column.c, 996
 - list, 663
- ewl_filelist_column_dir_change
 - Ewl_Filelist_Column, 191
- ewl_filelist_column_filename_get
 - Ewl_Filelist_Column, 191
- ewl_filelist_column_init
 - Ewl_Filelist_Column, 191
- EWL_FILELIST_COLUMN_IS
 - Ewl_Filelist_Column, 191
- ewl_filelist_column_new
 - Ewl_Filelist_Column, 192
- ewl_filelist_column_selected_file_add
 - Ewl_Filelist_Column, 192
- ewl_filelist_column_selected_unselect
 - Ewl_Filelist_Column, 192
- ewl_filelist_column_shift_handle
 - Ewl_Filelist_Column, 192
- EWL_FILELIST_COLUMN_TYPE
 - Ewl_Filelist_Column, 191
- ewl_filelist_column_view_get
 - Ewl_Filelist_Column, 192
- ewl_filelist_container_shift_handle
 - Ewl_Filelist, 183
- ewl_filelist_directory_get
 - Ewl_Filelist, 183
- ewl_filelist_directory_read
 - Ewl_Filelist, 183
- ewl_filelist_directory_set
 - Ewl_Filelist, 183
- EWL_FILELIST_EVENT_DIR_CHANGE
 - Ewl_Enums, 156
- EWL_FILELIST_EVENT_FILE_SELECTED
 - Ewl_Enums, 156
- EWL_FILELIST_EVENT_SELECTION_CHANGE
 - Ewl_Enums, 156
- Ewl_Filelist_Event_Type
 - Ewl_Enums, 151, 155
- ewl_filelist_expand_path
 - Ewl_Filelist, 183
- ewl_filelist_filter_get
 - Ewl_Filelist, 183
- ewl_filelist_filter_set
 - Ewl_Filelist, 184
- ewl_filelist_groupname_get
 - Ewl_Filelist, 184
- ewl_filelist_handle_click
 - Ewl_Filelist, 184
- ewl_filelist_hscroll_flag_get
 - Ewl_Filelist, 184
- ewl_filelist_hscroll_flag_set
 - Ewl_Filelist, 184
- EWL_FILELIST_ICON

- Ewl_Filelist_Icon, 194
- Ewl_Filelist_Icon, 664
- EWL_FILELIST_ICON, 194
- Ewl_Filelist_Icon, 194
- ewl_filelist_icon_dir_change, 194
- ewl_filelist_icon_filename_get, 194
- ewl_filelist_icon_init, 194
- EWL_FILELIST_ICON_IS, 194
- ewl_filelist_icon_new, 195
- ewl_filelist_icon_selected_file_add, 195
- ewl_filelist_icon_selected_unselect, 195
- ewl_filelist_icon_shift_handle, 195
- EWL_FILELIST_ICON_TYPE, 194
- ewl_filelist_icon_view_get, 195
- freebox, 664
- list, 664
- Ewl_Filelist_Icon: A icon file view, 193
- ewl_filelist_icon_dir_change
 - Ewl_Filelist_Icon, 194
- ewl_filelist_icon_filename_get
 - Ewl_Filelist_Icon, 194
- ewl_filelist_icon_init
 - Ewl_Filelist_Icon, 194
- EWL_FILELIST_ICON_IS
 - Ewl_Filelist_Icon, 194
- ewl_filelist_icon_new
 - Ewl_Filelist_Icon, 195
- ewl_filelist_icon_selected_file_add
 - Ewl_Filelist_Icon, 195
- ewl_filelist_icon_selected_unselect
 - Ewl_Filelist_Icon, 195
- ewl_filelist_icon_shift_handle
 - Ewl_Filelist_Icon, 195
- EWL_FILELIST_ICON_TYPE
 - Ewl_Filelist_Icon, 194
- ewl_filelist_icon_view_get
 - Ewl_Filelist_Icon, 195
- ewl_filelist_init
 - Ewl_Filelist, 185
- EWL_FILELIST_IS
 - Ewl_Filelist, 182
- EWL_FILELIST_LIST
 - Ewl_Filelist_List, 197
- Ewl_Filelist_List, 665
- EWL_FILELIST_LIST, 197
- Ewl_Filelist_List, 197
- ewl_filelist_list_dir_change, 197
- ewl_filelist_list_filename_get, 197
- ewl_filelist_list_init, 197
- EWL_FILELIST_LIST_IS, 197
- ewl_filelist_list_new, 198
- ewl_filelist_list_selected_file_add, 198
- ewl_filelist_list_selected_unselect, 198
- ewl_filelist_list_shift_handle, 198
- EWL_FILELIST_LIST_TYPE, 197
- ewl_filelist_list_view_get, 198
- list, 665
- tree, 665
- ewl_filelist_list.c
 - ewl_filelist_list_selected_file_add, 1003
- Ewl_Filelist_List: A listview of the files, 196
- ewl_filelist_list_dir_change
 - Ewl_Filelist_List, 197
- ewl_filelist_list_filename_get
 - Ewl_Filelist_List, 197
- ewl_filelist_list_init
 - Ewl_Filelist_List, 197
- EWL_FILELIST_LIST_IS
 - Ewl_Filelist_List, 197
- ewl_filelist_list_new
 - Ewl_Filelist_List, 198
- ewl_filelist_list_selected_file_add
 - Ewl_Filelist_List, 198
- ewl_filelist_list.c, 1003
- ewl_filelist_list_selected_unselect
 - Ewl_Filelist_List, 198
- ewl_filelist_list_shift_handle
 - Ewl_Filelist_List, 198
- EWL_FILELIST_LIST_TYPE
 - Ewl_Filelist_List, 197
- ewl_filelist_list_view_get
 - Ewl_Filelist_List, 198
- ewl_filelist_modtime_get
 - Ewl_Filelist, 185
- ewl_filelist_multiselect_get
 - Ewl_Filelist, 185
- ewl_filelist_multiselect_set
 - Ewl_Filelist, 185
- ewl_filelist_perms_get
 - Ewl_Filelist, 186
- ewl_filelist_selected_file_get
 - Ewl_Filelist, 186
- ewl_filelist_selected_file_preview_get
 - Ewl_Filelist, 186
- ewl_filelist_selected_file_set
 - Ewl_Filelist, 186
- ewl_filelist_selected_files_change_notify
 - Ewl_Filelist, 187
- ewl_filelist_selected_files_get
 - Ewl_Filelist, 187
- ewl_filelist_selected_files_set
 - Ewl_Filelist, 187
- ewl_filelist_selected_signal_all
 - Ewl_Filelist, 187
- ewl_filelist_show_dot_files_get
 - Ewl_Filelist, 188
- ewl_filelist_show_dot_files_set
 - Ewl_Filelist, 188

- ewl_filelist_size_get
 - Ewl_Filelist, 188
- ewl_filelist_stock_icon_get
 - Ewl_Filelist, 188
- EWL_FILELIST_TREE
 - Ewl_Filelist_Tree, 200
- Ewl_Filelist_Tree, 666
 - EWL_FILELIST_TREE, 200
 - Ewl_Filelist_Tree, 200
 - ewl_filelist_tree_dir_change, 200
 - ewl_filelist_tree_filename_get, 200
 - ewl_filelist_tree_init, 200
 - EWL_FILELIST_TREE_IS, 200
 - ewl_filelist_tree_new, 201
 - ewl_filelist_tree_selected_file_add, 201
 - ewl_filelist_tree_selected_unselect, 201
 - ewl_filelist_tree_shift_handle, 201
 - EWL_FILELIST_TREE_TYPE, 200
 - ewl_filelist_tree_view_get, 201
 - list, 666
 - tree, 666
- ewl_filelist_tree.c
 - EWL_FILELIST_TREE_DATA, 1007
 - Ewl_Filelist_Tree_Data, 1007
 - ewl_filelist_tree_selected_file_add, 1007
- Ewl_Filelist_Tree: A treeview of the files, 199
- EWL_FILELIST_TREE_DATA
 - ewl_filelist_tree.c, 1007
- Ewl_Filelist_Tree_Data, 667
 - ewl_filelist_tree.c, 1007
 - files, 667
 - list, 667
- ewl_filelist_tree_dir_change
 - Ewl_Filelist_Tree, 200
- ewl_filelist_tree_filename_get
 - Ewl_Filelist_Tree, 200
- ewl_filelist_tree_init
 - Ewl_Filelist_Tree, 200
- EWL_FILELIST_TREE_IS
 - Ewl_Filelist_Tree, 200
- ewl_filelist_tree_new
 - Ewl_Filelist_Tree, 201
- ewl_filelist_tree_selected_file_add
 - Ewl_Filelist_Tree, 201
 - ewl_filelist_tree.c, 1007
- ewl_filelist_tree_selected_unselect
 - Ewl_Filelist_Tree, 201
- ewl_filelist_tree_shift_handle
 - Ewl_Filelist_Tree, 201
- EWL_FILELIST_TREE_TYPE
 - Ewl_Filelist_Tree, 200
- ewl_filelist_tree_view_get
 - Ewl_Filelist_Tree, 201
- EWL_FILELIST_TYPE
 - Ewl_Filelist, 182
- ewl_filelist_username_get
 - Ewl_Filelist, 189
- ewl_filelist_vscroll_flag_get
 - Ewl_Filelist, 189
- ewl_filelist_vscroll_flag_set
 - Ewl_Filelist, 189
- EWL_FILEPICKER
 - Ewl_Filepicker, 204
- Ewl_Filepicker, 668
 - box, 668
 - EWL_FILEPICKER, 204
 - Ewl_Filepicker, 204
 - ewl_filepicker_directory_get, 205
 - ewl_filepicker_directory_set, 205
 - ewl_filepicker_filter_add, 205
 - ewl_filepicker_filter_get, 205
 - ewl_filepicker_filter_set, 205
 - ewl_filepicker_init, 206
 - EWL_FILEPICKER_IS, 204
 - ewl_filepicker_list_view_get, 206
 - ewl_filepicker_list_view_set, 206
 - ewl_filepicker_multiselect_get, 206
 - ewl_filepicker_multiselect_set, 207
 - ewl_filepicker_new, 207
 - ewl_filepicker_selected_file_get, 207
 - ewl_filepicker_selected_file_set, 207
 - ewl_filepicker_selected_files_get, 208
 - ewl_filepicker_selected_files_set, 208
 - ewl_filepicker_show_dot_files_get, 208
 - ewl_filepicker_show_dot_files_set, 208
 - ewl_filepicker_show_favorites_get, 209
 - ewl_filepicker_show_favorites_set, 209
 - EWL_FILEPICKER_TYPE, 204
 - favorites_box, 668
 - file_entry, 669
 - file_list, 669
 - file_list_box, 669
 - filters, 669
 - path, 669
 - path_combo, 669
 - show_favorites, 669
 - type_combo, 669
 - view, 669
- Ewl_Filepicker: Allows selecting files from a
 - list, 202
- ewl_filepicker_directory_get
 - Ewl_Filepicker, 205
- ewl_filepicker_directory_set
 - Ewl_Filepicker, 205
- Ewl_Filepicker_Filter, 670
 - filter, 670
 - lib/ewl_filepicker.c, 1012
 - name, 670

- ewl_filepicker_filter_add
 - Ewl_Filepicker, 205
- ewl_filepicker_filter_get
 - Ewl_Filepicker, 205
- ewl_filepicker_filter_set
 - Ewl_Filepicker, 205
- ewl_filepicker_init
 - Ewl_Filepicker, 206
- EWL_FILEPICKER_IS
 - Ewl_Filepicker, 204
- ewl_filepicker_list_view_get
 - Ewl_Filepicker, 206
- ewl_filepicker_list_view_set
 - Ewl_Filepicker, 206
- ewl_filepicker_multiselect_get
 - Ewl_Filepicker, 206
- ewl_filepicker_multiselect_set
 - Ewl_Filepicker, 207
- ewl_filepicker_new
 - Ewl_Filepicker, 207
- ewl_filepicker_selected_file_get
 - Ewl_Filepicker, 207
- ewl_filepicker_selected_file_set
 - Ewl_Filepicker, 207
- ewl_filepicker_selected_files_get
 - Ewl_Filepicker, 208
- ewl_filepicker_selected_files_set
 - Ewl_Filepicker, 208
- ewl_filepicker_show_dot_files_get
 - Ewl_Filepicker, 208
- ewl_filepicker_show_dot_files_set
 - Ewl_Filepicker, 208
- ewl_filepicker_show_favorites_get
 - Ewl_Filepicker, 209
- ewl_filepicker_show_favorites_set
 - Ewl_Filepicker, 209
- EWL_FILEPICKER_TYPE
 - Ewl_Filepicker, 204
- EWL_FLAG_ALIGN_BOTTOM
 - Ewl_Enums, 156
- EWL_FLAG_ALIGN_CENTER
 - Ewl_Enums, 156
- EWL_FLAG_ALIGN_LEFT
 - Ewl_Enums, 156
- EWL_FLAG_ALIGN_RIGHT
 - Ewl_Enums, 156
- EWL_FLAG_ALIGN_TOP
 - Ewl_Enums, 156
- EWL_FLAG_FILL_ALL
 - Ewl_Enums, 156
- EWL_FLAG_FILL_FILL
 - Ewl_Enums, 156
- EWL_FLAG_FILL_HFILL
 - Ewl_Enums, 156
- EWL_FLAG_FILL_HSHRINK
 - Ewl_Enums, 156
- EWL_FLAG_FILL_NONE
 - Ewl_Enums, 156
- EWL_FLAG_FILL_NORMAL
 - Ewl_Enums, 149
- EWL_FLAG_FILL_SHRINK
 - Ewl_Enums, 156
- EWL_FLAG_FILL_VFILL
 - Ewl_Enums, 156
- EWL_FLAG_FILL_VSHRINK
 - Ewl_Enums, 156
- EWL_FLAG_PROPERTY_BLOCK_TAB_ -
FOCUS
 - Ewl_Enums, 156
- EWL_FLAG_PROPERTY_DND_SOURCE
 - Ewl_Enums, 156
- EWL_FLAG_PROPERTY_DND_TARGET
 - Ewl_Enums, 156
- EWL_FLAG_PROPERTY_FOCUSABLE
 - Ewl_Enums, 156
- EWL_FLAG_PROPERTY_IN_TAB_LIST
 - Ewl_Enums, 156
- EWL_FLAG_PROPERTY_INTERNAL
 - Ewl_Enums, 156
- EWL_FLAG_PROPERTY_RECURSIVE
 - Ewl_Enums, 156
- EWL_FLAG_PROPERTY_TOPLEVEL
 - Ewl_Enums, 156
- EWL_FLAG_QUEUED_CPROCESS
 - Ewl_Enums, 157
- EWL_FLAG_QUEUED_CSCHEDULED
 - Ewl_Enums, 156
- EWL_FLAG_QUEUED_DPROCESS
 - Ewl_Enums, 157
- EWL_FLAG_QUEUED_DSCHEDULED
 - Ewl_Enums, 156
- EWL_FLAG_QUEUED_RPROCESS
 - Ewl_Enums, 157
- EWL_FLAG_QUEUED_RSCHEDULED
 - Ewl_Enums, 156
- EWL_FLAG_STATE_DISABLED
 - Ewl_Enums, 157
- EWL_FLAG_STATE_DND
 - Ewl_Enums, 157
- EWL_FLAG_STATE_DND_WAIT
 - Ewl_Enums, 157
- EWL_FLAG_STATE_FOCUSED
 - Ewl_Enums, 157
- EWL_FLAG_STATE_MOUSE_IN
 - Ewl_Enums, 157
- EWL_FLAG_STATE_NORMAL
 - Ewl_Enums, 157
- EWL_FLAG_STATE_PRESSED

- Ewl_Enums, 157
- EWL_FLAG_VISIBLE_HIDDEN
 - Ewl_Enums, 156
- EWL_FLAG_VISIBLE_NOCLIP
 - Ewl_Enums, 156
- EWL_FLAG_VISIBLE_OBSCURED
 - Ewl_Enums, 156
- EWL_FLAG_VISIBLE_REALIZED
 - Ewl_Enums, 156
- EWL_FLAG_VISIBLE_SHOWN
 - Ewl_Enums, 156
- Ewl_Flags
 - Ewl_Enums, 156
- EWL_FLAGS_ALIGN_MASK
 - Ewl_Enums, 149
- EWL_FLAGS_FILL_MASK
 - Ewl_Enums, 150
- EWL_FLAGS_PROPERTY_MASK
 - Ewl_Enums, 150
- EWL_FLAGS_QUEUED_MASK
 - Ewl_Enums, 150
- EWL_FLAGS_STATE_MASK
 - Ewl_Enums, 150
- EWL_FLAGS_VISIBLE_MASK
 - Ewl_Enums, 150
- EWL_FLOATER
 - Ewl_Floater, 211
- Ewl_Floater, 671
 - box, 671
 - EWL_FLOATER, 211
 - Ewl_Floater, 211
 - ewl_floater_cb_follow_configure, 211
 - ewl_floater_cb_follow_destroy, 211
 - ewl_floater_follow_get, 211
 - ewl_floater_follow_set, 211
 - ewl_floater_init, 212
 - EWL_FLOATER_IS, 211
 - ewl_floater_new, 212
 - ewl_floater_position_set, 212
 - EWL_FLOATER_TYPE, 211
 - follows, 671
 - x, 671
 - y, 671
- Ewl_Floater: A Floating Container, 210
- ewl_floater_cb_follow_configure
 - Ewl_Floater, 211
 - lib/ewl_floater.c, 1017
- ewl_floater_cb_follow_destroy
 - Ewl_Floater, 211
 - lib/ewl_floater.c, 1017
- ewl_floater_follow_get
 - Ewl_Floater, 211
- ewl_floater_follow_set
 - Ewl_Floater, 211
- ewl_floater_init
 - Ewl_Floater, 212
- EWL_FLOATER_IS
 - Ewl_Floater, 211
- ewl_floater_new
 - Ewl_Floater, 212
- ewl_floater_position_set
 - Ewl_Floater, 212
- ewl_floater_relative_set
 - lib/ewl_floater.c, 1017
- EWL_FLOATER_TYPE
 - Ewl_Floater, 211
- EWL_FREEBOX
 - Ewl_Freebox, 214
- Ewl_Freebox, 673
 - comparator, 673
 - container, 673
 - EWL_FREEBOX, 214
 - Ewl_Freebox, 214
 - ewl_freebox_cb_child_add, 215
 - ewl_freebox_cb_child_show, 215
 - ewl_freebox_cb_configure, 215
 - Ewl_Freebox_Comparator, 214
 - ewl_freebox_comparator_get, 215
 - ewl_freebox_comparator_set, 215
 - ewl_freebox_init, 215
 - EWL_FREEBOX_IS, 214
 - ewl_freebox_layout_type_get, 215
 - ewl_freebox_layout_type_set, 216
 - ewl_freebox_new, 216
 - ewl_freebox_orientation_get, 216
 - ewl_freebox_orientation_set, 216
 - ewl_freebox_resort, 217
 - EWL_FREEBOX_TYPE, 214
 - ewl_hfreebox_new, 217
 - ewl_vfreebox_new, 217
 - layout, 673
 - orientation, 673
 - sorted, 674
- Ewl_Freebox: A Freelayout widget, 213
- ewl_freebox_cb_child_add
 - Ewl_Freebox, 215
- ewl_freebox_cb_child_show
 - Ewl_Freebox, 215
- ewl_freebox_cb_configure
 - Ewl_Freebox, 215
 - lib/ewl_freebox.c, 1022
- Ewl_Freebox_Comparator
 - Ewl_Freebox, 214
- ewl_freebox_comparator_get
 - Ewl_Freebox, 215
- ewl_freebox_comparator_set
 - Ewl_Freebox, 215
- ewl_freebox_init

- Ewl_Freebox, 215
- EWL_FREEBOX_IS
 - Ewl_Freebox, 214
- EWL_FREEBOX_LAYOUT_AUTO
 - Ewl_Enums, 157
- EWL_FREEBOX_LAYOUT_-
 - COMPARATOR
 - Ewl_Enums, 157
- EWL_FREEBOX_LAYOUT_MANUAL
 - Ewl_Enums, 157
- Ewl_Freebox_Layout_Type
 - Ewl_Enums, 151, 157
- ewl_freebox_layout_type_get
 - Ewl_Freebox, 215
- ewl_freebox_layout_type_set
 - Ewl_Freebox, 216
- ewl_freebox_new
 - Ewl_Freebox, 216
- ewl_freebox_orientation_get
 - Ewl_Freebox, 216
- ewl_freebox_orientation_set
 - Ewl_Freebox, 216
- ewl_freebox_resort
 - Ewl_Freebox, 217
- EWL_FREEBOX_TYPE
 - Ewl_Freebox, 214
- ewl_fullscreen.c
 - test_info, 809
- EWL_GC_LIMIT
 - ewl_misc.c, 1101
- EWL_GRID
 - Ewl_Grid, 220
- Ewl_Grid, 675
 - col_size, 675
 - cols, 675
 - container, 675
 - data_dirty, 676
 - EWL_GRID, 220
 - Ewl_Grid, 220
 - ewl_grid_cb_child_add, 221
 - ewl_grid_cb_child_remove, 221
 - ewl_grid_cb_child_resize, 221
 - ewl_grid_cb_child_show, 221
 - ewl_grid_cb_configure, 221
 - ewl_grid_cb_destroy, 221
 - Ewl_Grid_Child, 220
 - ewl_grid_child_position_get, 221
 - ewl_grid_child_position_set, 221
 - ewl_grid_column_current_w_get, 222
 - ewl_grid_column_fixed_w_get, 222
 - ewl_grid_column_fixed_w_set, 222
 - ewl_grid_column_preferred_w_use, 222
 - ewl_grid_column_relative_w_get, 223
 - ewl_grid_column_relative_w_set, 223
 - ewl_grid_column_w_remove, 223
 - ewl_grid_dimensions_get, 223
 - ewl_grid_dimensions_set, 224
 - ewl_grid_hhomogeneous_get, 224
 - ewl_grid_hhomogeneous_set, 224
 - ewl_grid_homogeneous_set, 224
 - Ewl_Grid_Info, 221
 - ewl_grid_init, 225
 - EWL_GRID_IS, 220
 - ewl_grid_new, 225
 - ewl_grid_orientation_get, 225
 - ewl_grid_orientation_set, 225
 - ewl_grid_row_current_h_get, 226
 - ewl_grid_row_fixed_h_get, 226
 - ewl_grid_row_fixed_h_set, 226
 - ewl_grid_row_h_remove, 226
 - ewl_grid_row_preferred_h_use, 227
 - ewl_grid_row_relative_h_get, 227
 - ewl_grid_row_relative_h_set, 227
 - EWL_GRID_TYPE, 220
 - ewl_grid_vhomogeneous_get, 227
 - ewl_grid_vhomogeneous_set, 228
 - homogeneous_h, 676
 - homogeneous_v, 676
 - map, 676
 - orientation, 676
 - row_size, 676
 - rows, 676
 - space, 676
- Ewl_Grid The ewl grid widget, 218
- ewl_grid_cb_child_add
 - Ewl_Grid, 221
- ewl_grid_cb_child_remove
 - Ewl_Grid, 221
- ewl_grid_cb_child_resize
 - Ewl_Grid, 221
- ewl_grid_cb_child_show
 - Ewl_Grid, 221
- ewl_grid_cb_configure
 - Ewl_Grid, 221
 - lib/ewl_grid.c, 1028
- ewl_grid_cb_destroy
 - Ewl_Grid, 221
 - lib/ewl_grid.c, 1028
- Ewl_Grid_Child, 677
 - end_col, 677
 - end_row, 677
 - Ewl_Grid, 220
 - start_col, 677
 - start_row, 677
- ewl_grid_child_position_get
 - Ewl_Grid, 221
- ewl_grid_child_position_set
 - Ewl_Grid, 221

- ewl_grid_column_current_w_get
 - Ewl_Grid, 222
- ewl_grid_column_fixed_w_get
 - Ewl_Grid, 222
- ewl_grid_column_fixed_w_set
 - Ewl_Grid, 222
- ewl_grid_column_preferred_w_use
 - Ewl_Grid, 222
- ewl_grid_column_relative_w_get
 - Ewl_Grid, 223
- ewl_grid_column_relative_w_set
 - Ewl_Grid, 223
- ewl_grid_column_w_remove
 - Ewl_Grid, 223
- ewl_grid_dimensions_get
 - Ewl_Grid, 223
- ewl_grid_dimensions_set
 - Ewl_Grid, 224
- ewl_grid_hhomogeneous_get
 - Ewl_Grid, 224
- ewl_grid_hhomogeneous_set
 - Ewl_Grid, 224
- ewl_grid_homogeneous_set
 - Ewl_Grid, 224
- Ewl_Grid_Info, 678
 - current_pos, 678
 - current_size, 678
 - Ewl_Grid, 221
 - preferred_size, 678
 - rel_size, 678
 - resize_type, 678
 - size, 678
 - user, 678
- ewl_grid_init
 - Ewl_Grid, 225
- EWL_GRID_IS
 - Ewl_Grid, 220
- ewl_grid_new
 - Ewl_Grid, 225
- ewl_grid_orientation_get
 - Ewl_Grid, 225
- ewl_grid_orientation_set
 - Ewl_Grid, 225
- EWL_GRID_RESIZE_FIXED
 - Ewl_Enums, 157
- EWL_GRID_RESIZE_NONE
 - Ewl_Enums, 157
- EWL_GRID_RESIZE_NORMAL
 - Ewl_Enums, 157
- EWL_GRID_RESIZE_RELATIVE
 - Ewl_Enums, 157
- Ewl_Grid_Resize_Type
 - Ewl_Enums, 151, 157
- ewl_grid_row_current_h_get
 - Ewl_Grid, 226
- ewl_grid_row_fixed_h_get
 - Ewl_Grid, 226
- ewl_grid_row_fixed_h_set
 - Ewl_Grid, 226
- ewl_grid_row_h_remove
 - Ewl_Grid, 226
- ewl_grid_row_preferred_h_use
 - Ewl_Grid, 227
- ewl_grid_row_relative_h_get
 - Ewl_Grid, 227
- ewl_grid_row_relative_h_set
 - Ewl_Grid, 227
- EWL_GRID_TYPE
 - Ewl_Grid, 220
- ewl_grid_vhomogeneous_get
 - Ewl_Grid, 227
- ewl_grid_vhomogeneous_set
 - Ewl_Grid, 228
- ewl_hbox_new
 - Ewl_Box, 33
- ewl_hfreebox_new
 - Ewl_Freebox, 217
- EWL_HIGHLIGHT
 - Ewl_Highlight, 229
- Ewl_Highlight, 680
 - EWL_HIGHLIGHT, 229
 - Ewl_Highlight, 230
 - ewl_highlight_follow_get, 230
 - ewl_highlight_follow_set, 230
 - ewl_highlight_init, 230
 - EWL_HIGHLIGHT_IS, 229
 - ewl_highlight_new, 230
 - EWL_HIGHLIGHT_TYPE, 230
 - floater, 680
- Ewl_Highlight: A highlight widget, 229
- ewl_highlight_follow_get
 - Ewl_Highlight, 230
- ewl_highlight_follow_set
 - Ewl_Highlight, 230
- ewl_highlight_init
 - Ewl_Highlight, 230
- EWL_HIGHLIGHT_IS
 - Ewl_Highlight, 229
- ewl_highlight_new
 - Ewl_Highlight, 230
- EWL_HIGHLIGHT_TYPE
 - Ewl_Highlight, 230
- EWL_HISTOGRAM
 - Ewl_Histogram, 233
- Ewl_Histogram, 681
 - channel, 681
 - color, 681
 - EWL_HISTOGRAM, 233

- Ewl_Histogram, 233
- ewl_histogram_cb_configure, 233
- ewl_histogram_channel_get, 233
- ewl_histogram_channel_set, 233
- ewl_histogram_color_get, 234
- ewl_histogram_color_set, 234
- ewl_histogram_image_get, 234
- ewl_histogram_image_set, 235
- ewl_histogram_init, 235
- EWL_HISTOGRAM_IS, 233
- ewl_histogram_new, 235
- EWL_HISTOGRAM_TYPE, 233
- graph, 681
- image, 681
- maxv, 682
- source, 682
- Ewl_Histogram: A Simple Histogram widget, 232
- ewl_histogram_cb_configure
 - Ewl_Histogram, 233
 - lib/ewl_histogram.c, 1037
- Ewl_Histogram_Channel
 - Ewl_Enums, 151, 157
- EWL_HISTOGRAM_CHANNEL_B
 - Ewl_Enums, 157
- EWL_HISTOGRAM_CHANNEL_G
 - Ewl_Enums, 157
- ewl_histogram_channel_get
 - Ewl_Histogram, 233
- EWL_HISTOGRAM_CHANNEL_MAX
 - Ewl_Enums, 157
- EWL_HISTOGRAM_CHANNEL_R
 - Ewl_Enums, 157
- ewl_histogram_channel_set
 - Ewl_Histogram, 233
- EWL_HISTOGRAM_CHANNEL_Y
 - Ewl_Enums, 157
- ewl_histogram_color_get
 - Ewl_Histogram, 234
- ewl_histogram_color_set
 - Ewl_Histogram, 234
- ewl_histogram_image_get
 - Ewl_Histogram, 234
- ewl_histogram_image_set
 - Ewl_Histogram, 235
- ewl_histogram_init
 - Ewl_Histogram, 235
- EWL_HISTOGRAM_IS
 - Ewl_Histogram, 233
- ewl_histogram_new
 - Ewl_Histogram, 235
- EWL_HISTOGRAM_TYPE
 - Ewl_Histogram, 233
- ewl_hmenubar_new
 - Ewl_Menubar, 309
- ewl_hpaned_new
 - Ewl_Paned, 376
- ewl_hscrollbar_new
 - Ewl_Scrollbar, 405
- ewl_hseeker_new
 - Ewl_Seeker, 416
- ewl_hseparator_new
 - Ewl_Separator, 420
- ewl_htoolbar_new
 - Ewl_Toolbar, 491
- EWL_ICON
 - Ewl_Icon, 238
- Ewl_Icon, 683
 - alt, 684
 - complex_label, 684
 - compress_label, 684
 - editable, 684
 - EWL_ICON, 238
 - Ewl_Icon, 238
 - ewl_icon_alt_text_get, 238
 - ewl_icon_alt_text_set, 238
 - ewl_icon_cb_destroy, 239
 - ewl_icon_constrain_get, 239
 - ewl_icon_constrain_set, 239
 - ewl_icon_editable_get, 239
 - ewl_icon_editable_set, 239
 - ewl_icon_extended_data_get, 240
 - ewl_icon_extended_data_set, 240
 - ewl_icon_image_file_get, 240
 - ewl_icon_image_set, 240
 - ewl_icon_init, 241
 - EWL_ICON_IS, 238
 - ewl_icon_label_complex_get, 241
 - ewl_icon_label_complex_set, 241
 - ewl_icon_label_compressed_get, 241
 - ewl_icon_label_compressed_set, 242
 - ewl_icon_label_get, 242
 - ewl_icon_label_set, 242
 - ewl_icon_menu_get, 242
 - ewl_icon_menu_set, 243
 - ewl_icon_new, 243
 - ewl_icon_part_hide, 243
 - ewl_icon_simple_new, 243
 - ewl_icon_thumbnailing_get, 243
 - ewl_icon_thumbnailing_set, 244
 - EWL_ICON_TYPE, 238
 - ewl_icon_type_get, 244
 - ewl_icon_type_set, 244
 - extended, 684
 - hidden, 684
 - image, 684
 - label, 684
 - label_text, 684

- menu, 684
- stock, 684
- thumbnailing, 684
- type, 685
- Ewl_Icon: An icon widget, 236
- EWL_ICON_ACCESSORIES_-
CALCULATOR
 - Ewl_Icon_Theme, 251
- EWL_ICON_ACCESSORIES_-
CHARACTER_MAP
 - Ewl_Icon_Theme, 251
- EWL_ICON_ACCESSORIES_-
DICTIONARY
 - Ewl_Icon_Theme, 251
- EWL_ICON_ACCESSORIES_TEXT_-
EDITOR
 - Ewl_Icon_Theme, 251
- EWL_ICON_ADDRESS_BOOK_NEW
 - Ewl_Icon_Theme, 251
- ewl_icon_alt_text_get
 - Ewl_Icon, 238
- ewl_icon_alt_text_set
 - Ewl_Icon, 238
- EWL_ICON_APPLICATION_EXIT
 - Ewl_Icon_Theme, 251
- EWL_ICON_APPLICATION_X_-
EXECUTABLE
 - Ewl_Icon_Theme, 251
- EWL_ICON_APPLICATIONS_-
ACCESSORIES
 - Ewl_Icon_Theme, 251
- EWL_ICON_APPLICATIONS_-
DEVELOPMENT
 - Ewl_Icon_Theme, 252
- EWL_ICON_APPLICATIONS_GAMES
 - Ewl_Icon_Theme, 252
- EWL_ICON_APPLICATIONS_GRAPHICS
 - Ewl_Icon_Theme, 252
- EWL_ICON_APPLICATIONS_INTERNET
 - Ewl_Icon_Theme, 252
- EWL_ICON_APPLICATIONS_-
MULTIMEDIA
 - Ewl_Icon_Theme, 252
- EWL_ICON_APPLICATIONS_OFFICE
 - Ewl_Icon_Theme, 252
- EWL_ICON_APPLICATIONS_OTHER
 - Ewl_Icon_Theme, 252
- EWL_ICON_APPLICATIONS_SYSTEM
 - Ewl_Icon_Theme, 252
- EWL_ICON_APPLICATIONS_UTILITIES
 - Ewl_Icon_Theme, 252
- EWL_ICON_APPOINTMENT_MISSED
 - Ewl_Icon_Theme, 252
- EWL_ICON_APPOINTMENT_NEW
 - Ewl_Icon_Theme, 253
- EWL_ICON_APPOINTMENT_SOON
 - Ewl_Icon_Theme, 253
- EWL_ICON_AUDIO_CARD
 - Ewl_Icon_Theme, 253
- EWL_ICON_AUDIO_INPUT_-
MICROPHONE
 - Ewl_Icon_Theme, 253
- EWL_ICON_AUDIO_VOLUME_HIGH
 - Ewl_Icon_Theme, 253
- EWL_ICON_AUDIO_VOLUME_LOW
 - Ewl_Icon_Theme, 253
- EWL_ICON_AUDIO_VOLUME_MEDIUM
 - Ewl_Icon_Theme, 253
- EWL_ICON_AUDIO_VOLUME_MUTED
 - Ewl_Icon_Theme, 253
- EWL_ICON_AUDIO_X_GENERIC
 - Ewl_Icon_Theme, 253
- EWL_ICON_BATTERY
 - Ewl_Icon_Theme, 253
- EWL_ICON_BATTERY_CAUTION
 - Ewl_Icon_Theme, 253
- EWL_ICON_BATTERY_LOW
 - Ewl_Icon_Theme, 254
- EWL_ICON_CAMERA_PHOTO
 - Ewl_Icon_Theme, 254
- EWL_ICON_CAMERA_VIDEO
 - Ewl_Icon_Theme, 254
- ewl_icon_cb_destroy
 - Ewl_Icon, 239
- lib/ewl_icon.c, 1043
- EWL_ICON_COMPRESS_SIZE
 - lib/ewl_icon.c, 1043
- EWL_ICON_COMPUTER
 - Ewl_Icon_Theme, 254
- ewl_icon_constrain_get
 - Ewl_Icon, 239
- ewl_icon_constrain_set
 - Ewl_Icon, 239
- EWL_ICON_CONTACT_NEW
 - Ewl_Icon_Theme, 254
- EWL_ICON_DIALOG_APPLY
 - Ewl_Icon_Theme, 254
- EWL_ICON_DIALOG_CANCEL
 - Ewl_Icon_Theme, 254
- EWL_ICON_DIALOG_CLOSE
 - Ewl_Icon_Theme, 254
- EWL_ICON_DIALOG_ERROR
 - Ewl_Icon_Theme, 254
- EWL_ICON_DIALOG_INFORMATION
 - Ewl_Icon_Theme, 254
- EWL_ICON_DIALOG_OK
 - Ewl_Icon_Theme, 254
- EWL_ICON_DIALOG_PASSWORD

- Ewl_Icon_Theme, 254
- EWL_ICON_DIALOG_QUESTION
 - Ewl_Icon_Theme, 255
- EWL_ICON_DIALOG_WARNING
 - Ewl_Icon_Theme, 255
- EWL_ICON_DOCUMENT_NEW
 - Ewl_Icon_Theme, 255
- EWL_ICON_DOCUMENT_OPEN
 - Ewl_Icon_Theme, 255
- EWL_ICON_DOCUMENT_OPEN_-
RECENT
 - Ewl_Icon_Theme, 255
- EWL_ICON_DOCUMENT_PAGE_SETUP
 - Ewl_Icon_Theme, 255
- EWL_ICON_DOCUMENT_PRINT
 - Ewl_Icon_Theme, 255
- EWL_ICON_DOCUMENT_PRINT_-
PREVIEW
 - Ewl_Icon_Theme, 255
- EWL_ICON_DOCUMENT_PROPERTIES
 - Ewl_Icon_Theme, 255
- EWL_ICON_DOCUMENT_REVERT
 - Ewl_Icon_Theme, 255
- EWL_ICON_DOCUMENT_SAVE
 - Ewl_Icon_Theme, 255
- EWL_ICON_DOCUMENT_SAVE_AS
 - Ewl_Icon_Theme, 256
- EWL_ICON_DRIVE_CDROM
 - Ewl_Icon_Theme, 256
- EWL_ICON_DRIVE_HARDDISK
 - Ewl_Icon_Theme, 256
- EWL_ICON_DRIVE_REMOVABLE_-
MEDIA
 - Ewl_Icon_Theme, 256
- EWL_ICON_EDIT_COPY
 - Ewl_Icon_Theme, 256
- EWL_ICON_EDIT_CUT
 - Ewl_Icon_Theme, 256
- EWL_ICON_EDIT_DELETE
 - Ewl_Icon_Theme, 256
- EWL_ICON_EDIT_FIND
 - Ewl_Icon_Theme, 256
- EWL_ICON_EDIT_FIND_REPLACE
 - Ewl_Icon_Theme, 256
- EWL_ICON_EDIT_PASTE
 - Ewl_Icon_Theme, 256
- EWL_ICON_EDIT_REDO
 - Ewl_Icon_Theme, 256
- EWL_ICON_EDIT_SELECT_ALL
 - Ewl_Icon_Theme, 257
- EWL_ICON_EDIT_UNDO
 - Ewl_Icon_Theme, 257
- ewl_icon_editable_get
 - Ewl_Icon, 239
- ewl_icon_editable_set
 - Ewl_Icon, 239
- EWL_ICON_EMBLEM_DEFAULT
 - Ewl_Icon_Theme, 257
- EWL_ICON_EMBLEM_DOCUMENTS
 - Ewl_Icon_Theme, 257
- EWL_ICON_EMBLEM_DOWNLOADS
 - Ewl_Icon_Theme, 257
- EWL_ICON_EMBLEM_FAVORITE
 - Ewl_Icon_Theme, 257
- EWL_ICON_EMBLEM_IMPORTANT
 - Ewl_Icon_Theme, 257
- EWL_ICON_EMBLEM_MAIL
 - Ewl_Icon_Theme, 257
- EWL_ICON_EMBLEM_PHOTOS
 - Ewl_Icon_Theme, 257
- EWL_ICON_EMBLEM_READONLY
 - Ewl_Icon_Theme, 257
- EWL_ICON_EMBLEM_SHARED
 - Ewl_Icon_Theme, 257
- EWL_ICON_EMBLEM_SYMBOLIC_LINK
 - Ewl_Icon_Theme, 257
- EWL_ICON_EMBLEM_SYNCHRONIZED
 - Ewl_Icon_Theme, 258
- EWL_ICON_EMBLEM_SYSTEM
 - Ewl_Icon_Theme, 258
- EWL_ICON_EMBLEM_UNREADABLE
 - Ewl_Icon_Theme, 258
- ewl_icon_extended_data_get
 - Ewl_Icon, 240
- ewl_icon_extended_data_set
 - Ewl_Icon, 240
- EWL_ICON_FACE_ANGEL
 - Ewl_Icon_Theme, 258
- EWL_ICON_FACE_CRYING
 - Ewl_Icon_Theme, 258
- EWL_ICON_FACE_DEVIL_GRIN
 - Ewl_Icon_Theme, 258
- EWL_ICON_FACE_DEVIL_SAD
 - Ewl_Icon_Theme, 258
- EWL_ICON_FACE_GLASSES
 - Ewl_Icon_Theme, 258
- EWL_ICON_FACE_KISS
 - Ewl_Icon_Theme, 258
- EWL_ICON_FACE_MONKEY
 - Ewl_Icon_Theme, 258
- EWL_ICON_FACE_PLAIN
 - Ewl_Icon_Theme, 258
- EWL_ICON_FACE_SAD
 - Ewl_Icon_Theme, 259
- EWL_ICON_FACE_SMILE
 - Ewl_Icon_Theme, 259
- EWL_ICON_FACE_SMILE_BIG
 - Ewl_Icon_Theme, 259

- EWL_ICON_FACE_SMIRK
 - Ewl_Icon_Theme, 259
- EWL_ICON_FACE_SURPRISE
 - Ewl_Icon_Theme, 259
- EWL_ICON_FACE_WINK
 - Ewl_Icon_Theme, 259
- EWL_ICON_FOLDER
 - Ewl_Icon_Theme, 259
- EWL_ICON_FOLDER_DRAG_ACCEPT
 - Ewl_Icon_Theme, 259
- EWL_ICON_FOLDER_OPEN
 - Ewl_Icon_Theme, 259
- EWL_ICON_FOLDER_REMOTE
 - Ewl_Icon_Theme, 259
- EWL_ICON_FOLDER_VISITING
 - Ewl_Icon_Theme, 259
- EWL_ICON_FONT_X_GENERIC
 - Ewl_Icon_Theme, 260
- EWL_ICON_FORMAT_INDENT_LESS
 - Ewl_Icon_Theme, 260
- EWL_ICON_FORMAT_INDENT_MORE
 - Ewl_Icon_Theme, 260
- EWL_ICON_FORMAT_JUSTIFY_-
 - CENTER
 - Ewl_Icon_Theme, 260
- EWL_ICON_FORMAT_JUSTIFY_FILL
 - Ewl_Icon_Theme, 260
- EWL_ICON_FORMAT_JUSTIFY_LEFT
 - Ewl_Icon_Theme, 260
- EWL_ICON_FORMAT_JUSTIFY_RIGHT
 - Ewl_Icon_Theme, 260
- EWL_ICON_FORMAT_TEXT_BOLD
 - Ewl_Icon_Theme, 260
- EWL_ICON_FORMAT_TEXT_-
 - DIRECTION_LTR
 - Ewl_Icon_Theme, 260
 - DIRECTION_RTL
 - Ewl_Icon_Theme, 260
- EWL_ICON_FORMAT_TEXT_ITALIC
 - Ewl_Icon_Theme, 261
- EWL_ICON_FORMAT_TEXT_-
 - STRIKETHROUGH
 - Ewl_Icon_Theme, 261
 - UNDERLINE
 - Ewl_Icon_Theme, 261
- EWL_ICON_GO_BOTTOM
 - Ewl_Icon_Theme, 261
- EWL_ICON_GO_DOWN
 - Ewl_Icon_Theme, 261
- EWL_ICON_GO_FIRST
 - Ewl_Icon_Theme, 261
- EWL_ICON_GO_HOME
 - Ewl_Icon_Theme, 261
- EWL_ICON_GO_JUMP
 - Ewl_Icon_Theme, 261
- EWL_ICON_GO_LAST
 - Ewl_Icon_Theme, 261
- EWL_ICON_GO_NEXT
 - Ewl_Icon_Theme, 261
- EWL_ICON_GO_PREVIOUS
 - Ewl_Icon_Theme, 261
- EWL_ICON_GO_TOP
 - Ewl_Icon_Theme, 262
- EWL_ICON_GO_UP
 - Ewl_Icon_Theme, 262
- EWL_ICON_HELP_ABOUT
 - Ewl_Icon_Theme, 262
- EWL_ICON_HELP_BROWSER
 - Ewl_Icon_Theme, 262
- EWL_ICON_HELP_CONTENTS
 - Ewl_Icon_Theme, 262
- EWL_ICON_HELP_FAQ
 - Ewl_Icon_Theme, 262
- ewl_icon_image_file_get
 - Ewl_Icon, 240
- EWL_ICON_IMAGE_LOADING
 - Ewl_Icon_Theme, 262
- EWL_ICON_IMAGE_MISSING
 - Ewl_Icon_Theme, 262
- ewl_icon_image_set
 - Ewl_Icon, 240
- EWL_ICON_IMAGE_X_GENERIC
 - Ewl_Icon_Theme, 262
- ewl_icon_init
 - Ewl_Icon, 241
- EWL_ICON_INPUT_GAMING
 - Ewl_Icon_Theme, 262
- EWL_ICON_INPUT_KEYBOARD
 - Ewl_Icon_Theme, 262
- EWL_ICON_INPUT_MOUSE
 - Ewl_Icon_Theme, 262
- EWL_ICON_INSERT_IMAGE
 - Ewl_Icon_Theme, 263
- EWL_ICON_INSERT_LINK
 - Ewl_Icon_Theme, 263
- EWL_ICON_INSERT_OBJECT
 - Ewl_Icon_Theme, 263
- EWL_ICON_INSERT_TEXT
 - Ewl_Icon_Theme, 263
- EWL_ICON_IS
 - Ewl_Icon, 238
- ewl_icon_label_complex_get
 - Ewl_Icon, 241
- ewl_icon_label_complex_set
 - Ewl_Icon, 241
- ewl_icon_label_compressed_get

- Ewl_Icon, 241
- ewl_icon_label_compressed_set
 - Ewl_Icon, 242
- ewl_icon_label_get
 - Ewl_Icon, 242
- ewl_icon_label_set
 - Ewl_Icon, 242
- EWL_ICON_LIST_ADD
 - Ewl_Icon_Theme, 263
- EWL_ICON_LIST_REMOVE
 - Ewl_Icon_Theme, 263
- EWL_ICON_MAIL_ATTACHMENT
 - Ewl_Icon_Theme, 263
- EWL_ICON_MAIL_FORWARD
 - Ewl_Icon_Theme, 263
- EWL_ICON_MAIL_MARK_IMPORTANT
 - Ewl_Icon_Theme, 263
- EWL_ICON_MAIL_MARK_JUNK
 - Ewl_Icon_Theme, 263
- EWL_ICON_MAIL_MARK_NOTJUNK
 - Ewl_Icon_Theme, 263
- EWL_ICON_MAIL_MARK_READ
 - Ewl_Icon_Theme, 264
- EWL_ICON_MAIL_MARK_UNREAD
 - Ewl_Icon_Theme, 264
- EWL_ICON_MAIL_MESSAGE_NEW
 - Ewl_Icon_Theme, 264
- EWL_ICON_MAIL_READ
 - Ewl_Icon_Theme, 264
- EWL_ICON_MAIL_REPLIED
 - Ewl_Icon_Theme, 264
- EWL_ICON_MAIL_REPLY_ALL
 - Ewl_Icon_Theme, 264
- EWL_ICON_MAIL_REPLY_SENDER
 - Ewl_Icon_Theme, 264
- EWL_ICON_MAIL_SEND_RECEIVE
 - Ewl_Icon_Theme, 264
- EWL_ICON_MAIL_SIGNED
 - Ewl_Icon_Theme, 264
- EWL_ICON_MAIL_SIGNED_VERIFIED
 - Ewl_Icon_Theme, 264
- EWL_ICON_MAIL_UNREAD
 - Ewl_Icon_Theme, 264
- EWL_ICON_MEDIA_CDROM
 - Ewl_Icon_Theme, 265
- EWL_ICON_MEDIA_EJECT
 - Ewl_Icon_Theme, 265
- EWL_ICON_MEDIA_FLOPPY
 - Ewl_Icon_Theme, 265
- EWL_ICON_MEDIA_PLAYBACK_PAUSE
 - Ewl_Icon_Theme, 265
- EWL_ICON_MEDIA_PLAYBACK_START
 - Ewl_Icon_Theme, 265
- EWL_ICON_MEDIA_PLAYBACK_STOP
 - Ewl_Icon_Theme, 265
- EWL_ICON_MEDIA_PLAYLIST_REPEAT
 - Ewl_Icon_Theme, 265
- EWL_ICON_MEDIA_PLAYLIST_SHUFFLE
 - Ewl_Icon_Theme, 265
- EWL_ICON_MEDIA_RECORD
 - Ewl_Icon_Theme, 265
- EWL_ICON_MEDIA_SEEK_BACKWARD
 - Ewl_Icon_Theme, 265
- EWL_ICON_MEDIA_SEEK_FORWARD
 - Ewl_Icon_Theme, 266
- EWL_ICON_MEDIA_SKIP_BACKWARD
 - Ewl_Icon_Theme, 266
- EWL_ICON_MEDIA_SKIP_FORWARD
 - Ewl_Icon_Theme, 266
- ewl_icon_menu_get
 - Ewl_Icon, 242
- ewl_icon_menu_set
 - Ewl_Icon, 243
- EWL_ICON_MULTIMEDIA_PLAYER
 - Ewl_Icon_Theme, 266
- EWL_ICON_MULTIMEDIA_VOLUME_CONTROL
 - Ewl_Icon_Theme, 266
- EWL_ICON_NETWORK_ERROR
 - Ewl_Icon_Theme, 266
- EWL_ICON_NETWORK_IDLE
 - Ewl_Icon_Theme, 266
- EWL_ICON_NETWORK_OFFLINE
 - Ewl_Icon_Theme, 266
- EWL_ICON_NETWORK_RECEIVE
 - Ewl_Icon_Theme, 266
- EWL_ICON_NETWORK_SERVER
 - Ewl_Icon_Theme, 266
- EWL_ICON_NETWORK_TRANSMIT
 - Ewl_Icon_Theme, 266
- EWL_ICON_NETWORK_TRANSMIT_RECEIVE
 - Ewl_Icon_Theme, 267
- EWL_ICON_NETWORK_WIRED
 - Ewl_Icon_Theme, 267
- EWL_ICON_NETWORK_WIRELESS
 - Ewl_Icon_Theme, 267
- EWL_ICON_NETWORK_WORKGROUP
 - Ewl_Icon_Theme, 267
- ewl_icon_new
 - Ewl_Icon, 243
- EWL_ICON_PACKAGE_X_GENERIC
 - Ewl_Icon_Theme, 267
- Ewl_Icon_Part
 - Ewl_Enums, 152, 157
- ewl_icon_part_hide
 - Ewl_Icon, 243

- EWL_ICON_PART_IMAGE
 - Ewl_Enums, 158
- EWL_ICON_PART_LABEL
 - Ewl_Enums, 158
- EWL_ICON_PART_NONE
 - Ewl_Enums, 158
- EWL_ICON_PREFERENCES_DESKTOP
 - Ewl_Icon_Theme, 267
- EWL_ICON_PREFERENCES_-
 - DESKTOP_ACCESSIBILITY
 - Ewl_Icon_Theme, 267
- EWL_ICON_PREFERENCES_-
 - DESKTOP_FONT
 - Ewl_Icon_Theme, 267
- EWL_ICON_PREFERENCES_-
 - DESKTOP_KEYBOARD
 - Ewl_Icon_Theme, 267
- EWL_ICON_PREFERENCES_-
 - DESKTOP_LOCALE
 - Ewl_Icon_Theme, 268
- EWL_ICON_PREFERENCES_-
 - DESKTOP_MULTIMEDIA
 - Ewl_Icon_Theme, 268
- EWL_ICON_PREFERENCES_-
 - DESKTOP_PERIPHERALS
 - Ewl_Icon_Theme, 268
- EWL_ICON_PREFERENCES_-
 - DESKTOP_PERSONAL
 - Ewl_Icon_Theme, 268
- EWL_ICON_PREFERENCES_-
 - DESKTOP_SCREENSAVER
 - Ewl_Icon_Theme, 268
- EWL_ICON_PREFERENCES_-
 - DESKTOP_THEME
 - Ewl_Icon_Theme, 268
- EWL_ICON_PREFERENCES_-
 - DESKTOP_WALLPAPER
 - Ewl_Icon_Theme, 268
- EWL_ICON_PREFERENCES_OTHER
 - Ewl_Icon_Theme, 268
- EWL_ICON_PREFERENCES_SYSTEM
 - Ewl_Icon_Theme, 268
- EWL_ICON_PREFERENCES_SYSTEM_-
 - NETWORK
 - Ewl_Icon_Theme, 268
- EWL_ICON_PRINTER
 - Ewl_Icon_Theme, 269
- EWL_ICON_PRINTER_ERROR
 - Ewl_Icon_Theme, 269
- EWL_ICON_PRINTER_PRINTING
 - Ewl_Icon_Theme, 269
- EWL_ICON_PROCESS_WORKING
 - Ewl_Icon_Theme, 269
- ewl_icon_simple_new
 - Ewl_Icon, 243
- EWL_ICON_SIZE_LARGE
 - Ewl_Icon_Theme, 269
- EWL_ICON_SIZE_MEDIUM
 - Ewl_Icon_Theme, 269
- EWL_ICON_SIZE_SMALL
 - Ewl_Icon_Theme, 269
- EWL_ICON_SOFTWARE_UPDATE_-
 - AVAILABLE
 - Ewl_Icon_Theme, 269
- EWL_ICON_SOFTWARE_UPDATE_-
 - URGENT
 - Ewl_Icon_Theme, 269
- EWL_ICON_START_HERE
 - Ewl_Icon_Theme, 269
- EWL_ICON_SYNC_ERROR
 - Ewl_Icon_Theme, 269
- EWL_ICON_SYNC_SYNCHRONIZING
 - Ewl_Icon_Theme, 270
- EWL_ICON_SYSTEM_FILE_MANAGER
 - Ewl_Icon_Theme, 270
- EWL_ICON_SYSTEM_HELP
 - Ewl_Icon_Theme, 270
- EWL_ICON_SYSTEM_LOCK_SCREEN
 - Ewl_Icon_Theme, 270
- EWL_ICON_SYSTEM_LOG_OUT
 - Ewl_Icon_Theme, 270
- EWL_ICON_SYSTEM_RUN
 - Ewl_Icon_Theme, 270
- EWL_ICON_SYSTEM_SEARCH
 - Ewl_Icon_Theme, 270
- EWL_ICON_SYSTEM_SOFTWARE_-
 - UPDATE
 - Ewl_Icon_Theme, 270
- EWL_ICON_TASK_DUE
 - Ewl_Icon_Theme, 270
- EWL_ICON_TASK_PASSED_DUE
 - Ewl_Icon_Theme, 270
- EWL_ICON_TEXT_HTML
 - Ewl_Icon_Theme, 270
- EWL_ICON_TEXT_X_GENERIC
 - Ewl_Icon_Theme, 271
- EWL_ICON_TEXT_X_GENERIC_-
 - TEMPLATE
 - Ewl_Icon_Theme, 271
- EWL_ICON_TEXT_X_SCRIPT
 - Ewl_Icon_Theme, 271
- Ewl_Icon_Theme
 - EWL_ICON_ACCESSORIES_-
 - CALCULATOR, 251
 - EWL_ICON_ACCESSORIES_-
 - CHARACTER_MAP, 251
 - EWL_ICON_ACCESSORIES_-
 - DICTIONARY, 251

- EWL_ICON_ACCESSORIES_TEXT _ -
EDITOR, 251
- EWL_ICON_ADDRESS_BOOK_NEW,
251
- EWL_ICON_APPLICATION_EXIT,
251
- EWL_ICON_APPLICATION_X _ -
EXECUTABLE, 251
- EWL_ICON_APPLICATIONS _ -
ACCESSORIES, 251
- EWL_ICON_APPLICATIONS _ -
DEVELOPMENT, 252
- EWL_ICON_APPLICATIONS _ -
GAMES, 252
- EWL_ICON_APPLICATIONS _ -
GRAPHICS, 252
- EWL_ICON_APPLICATIONS _ -
INTERNET, 252
- EWL_ICON_APPLICATIONS _ -
MULTIMEDIA, 252
- EWL_ICON_APPLICATIONS _ -
OFFICE, 252
- EWL_ICON_APPLICATIONS _ -
OTHER, 252
- EWL_ICON_APPLICATIONS _ -
SYSTEM, 252
- EWL_ICON_APPLICATIONS _ -
UTILITIES, 252
- EWL_ICON_APPOINTMENT _ -
MISSED, 252
- EWL_ICON_APPOINTMENT_NEW,
253
- EWL_ICON_APPOINTMENT_SOON,
253
- EWL_ICON_AUDIO_CARD, 253
- EWL_ICON_AUDIO_INPUT _ -
MICROPHONE, 253
- EWL_ICON_AUDIO_VOLUME _ -
HIGH, 253
- EWL_ICON_AUDIO_VOLUME_LOW,
253
- EWL_ICON_AUDIO_VOLUME _ -
MEDIUM, 253
- EWL_ICON_AUDIO_VOLUME _ -
MUTED, 253
- EWL_ICON_AUDIO_X_GENERIC,
253
- EWL_ICON_BATTERY, 253
- EWL_ICON_BATTERY_CAUTION,
253
- EWL_ICON_BATTERY_LOW, 254
- EWL_ICON_CAMERA_PHOTO, 254
- EWL_ICON_CAMERA_VIDEO, 254
- EWL_ICON_COMPUTER, 254
- EWL_ICON_CONTACT_NEW, 254
- EWL_ICON_DIALOG_APPLY, 254
- EWL_ICON_DIALOG_CANCEL, 254
- EWL_ICON_DIALOG_CLOSE, 254
- EWL_ICON_DIALOG_ERROR, 254
- EWL_ICON_DIALOG _ -
INFORMATION, 254
- EWL_ICON_DIALOG_OK, 254
- EWL_ICON_DIALOG_PASSWORD,
254
- EWL_ICON_DIALOG_QUESTION, 255
- EWL_ICON_DIALOG_WARNING, 255
- EWL_ICON_DOCUMENT_NEW, 255
- EWL_ICON_DOCUMENT_OPEN, 255
- EWL_ICON_DOCUMENT_OPEN _ -
RECENT, 255
- EWL_ICON_DOCUMENT_PAGE _ -
SETUP, 255
- EWL_ICON_DOCUMENT_PRINT, 255
- EWL_ICON_DOCUMENT_PRINT _ -
PREVIEW, 255
- EWL_ICON_DOCUMENT _ -
PROPERTIES, 255
- EWL_ICON_DOCUMENT_REVERT,
255
- EWL_ICON_DOCUMENT_SAVE, 255
- EWL_ICON_DOCUMENT_SAVE_AS,
256
- EWL_ICON_DRIVE_CDROM, 256
- EWL_ICON_DRIVE_HARDDISK, 256
- EWL_ICON_DRIVE_REMOVABLE _ -
MEDIA, 256
- EWL_ICON_EDIT_COPY, 256
- EWL_ICON_EDIT_CUT, 256
- EWL_ICON_EDIT_DELETE, 256
- EWL_ICON_EDIT_FIND, 256
- EWL_ICON_EDIT_FIND_REPLACE,
256
- EWL_ICON_EDIT_PASTE, 256
- EWL_ICON_EDIT_REDO, 256
- EWL_ICON_EDIT_SELECT_ALL, 257
- EWL_ICON_EDIT_UNDO, 257
- EWL_ICON_EMBLEM_DEFAULT, 257
- EWL_ICON_EMBLEM_DOCUMENTS,
257
- EWL_ICON_EMBLEM _ -
DOWNLOADS, 257
- EWL_ICON_EMBLEM_FAVORITE,
257
- EWL_ICON_EMBLEM_IMPORTANT,
257
- EWL_ICON_EMBLEM_MAIL, 257
- EWL_ICON_EMBLEM_PHOTOS, 257

- EWL_ICON_EMBLEM_READONLY, 257
- EWL_ICON_EMBLEM_SHARED, 257
- EWL_ICON_EMBLEM_SYMBOLIC_LINK, 257
- EWL_ICON_EMBLEM_-
SYNCHRONIZED, 258
- EWL_ICON_EMBLEM_SYSTEM, 258
- EWL_ICON_EMBLEM_-
UNREADABLE, 258
- EWL_ICON_FACE_ANGEL, 258
- EWL_ICON_FACE_CRYING, 258
- EWL_ICON_FACE_DEVIL_GRIN, 258
- EWL_ICON_FACE_DEVIL_SAD, 258
- EWL_ICON_FACE_GLASSES, 258
- EWL_ICON_FACE_KISS, 258
- EWL_ICON_FACE_MONKEY, 258
- EWL_ICON_FACE_PLAIN, 258
- EWL_ICON_FACE_SAD, 259
- EWL_ICON_FACE_SMILE, 259
- EWL_ICON_FACE_SMILE_BIG, 259
- EWL_ICON_FACE_SMIRK, 259
- EWL_ICON_FACE_SURPRISE, 259
- EWL_ICON_FACE_WINK, 259
- EWL_ICON_FOLDER, 259
- EWL_ICON_FOLDER_DRAG_-
ACCEPT, 259
- EWL_ICON_FOLDER_OPEN, 259
- EWL_ICON_FOLDER_REMOTE, 259
- EWL_ICON_FOLDER_VISITING, 259
- EWL_ICON_FONT_X_GENERIC, 260
- EWL_ICON_FORMAT_INDENT_-
LESS, 260
- EWL_ICON_FORMAT_INDENT_-
MORE, 260
- EWL_ICON_FORMAT_JUSTIFY_-
CENTER, 260
- EWL_ICON_FORMAT_JUSTIFY_-
FILL, 260
- EWL_ICON_FORMAT_JUSTIFY_-
LEFT, 260
- EWL_ICON_FORMAT_JUSTIFY_-
RIGHT, 260
- EWL_ICON_FORMAT_TEXT_BOLD, 260
- EWL_ICON_FORMAT_TEXT_-
DIRECTION_LTR, 260
- EWL_ICON_FORMAT_TEXT_-
DIRECTION_RTL, 260
- EWL_ICON_FORMAT_TEXT_-
ITALIC, 261
- EWL_ICON_FORMAT_TEXT_-
STRIKETHROUGH, 261
- EWL_ICON_FORMAT_TEXT_-
UNDERLINE, 261
- EWL_ICON_GO_BOTTOM, 261
- EWL_ICON_GO_DOWN, 261
- EWL_ICON_GO_FIRST, 261
- EWL_ICON_GO_HOME, 261
- EWL_ICON_GO_JUMP, 261
- EWL_ICON_GO_LAST, 261
- EWL_ICON_GO_NEXT, 261
- EWL_ICON_GO_PREVIOUS, 261
- EWL_ICON_GO_TOP, 262
- EWL_ICON_GO_UP, 262
- EWL_ICON_HELP_ABOUT, 262
- EWL_ICON_HELP_BROWSER, 262
- EWL_ICON_HELP_CONTENTS, 262
- EWL_ICON_HELP_FAQ, 262
- EWL_ICON_IMAGE_LOADING, 262
- EWL_ICON_IMAGE_MISSING, 262
- EWL_ICON_IMAGE_X_GENERIC, 262
- EWL_ICON_INPUT_GAMING, 262
- EWL_ICON_INPUT_KEYBOARD, 262
- EWL_ICON_INPUT_MOUSE, 262
- EWL_ICON_INSERT_IMAGE, 263
- EWL_ICON_INSERT_LINK, 263
- EWL_ICON_INSERT_OBJECT, 263
- EWL_ICON_INSERT_TEXT, 263
- EWL_ICON_LIST_ADD, 263
- EWL_ICON_LIST_REMOVE, 263
- EWL_ICON_MAIL_ATTACHMENT, 263
- EWL_ICON_MAIL_FORWARD, 263
- EWL_ICON_MAIL_MARK_-
IMPORTANT, 263
- EWL_ICON_MAIL_MARK_JUNK, 263
- EWL_ICON_MAIL_MARK_-
NOTJUNK, 263
- EWL_ICON_MAIL_MARK_READ, 264
- EWL_ICON_MAIL_MARK_UNREAD, 264
- EWL_ICON_MAIL_MESSAGE_NEW, 264
- EWL_ICON_MAIL_READ, 264
- EWL_ICON_MAIL_REPLIED, 264
- EWL_ICON_MAIL_REPLY_ALL, 264
- EWL_ICON_MAIL_REPLY_SENDER, 264
- EWL_ICON_MAIL_SEND_RECEIVE, 264
- EWL_ICON_MAIL_SIGNED, 264
- EWL_ICON_MAIL_SIGNED_-
VERIFIED, 264
- EWL_ICON_MAIL_UNREAD, 264

- EWL_ICON_MEDIA_CDROM, 265
- EWL_ICON_MEDIA_EJECT, 265
- EWL_ICON_MEDIA_FLOPPY, 265
- EWL_ICON_MEDIA_PLAYBACK_-
PAUSE, 265
- EWL_ICON_MEDIA_PLAYBACK_-
START, 265
- EWL_ICON_MEDIA_PLAYBACK_-
STOP, 265
- EWL_ICON_MEDIA_PLAYLIST_-
REPEAT, 265
- EWL_ICON_MEDIA_PLAYLIST_-
SHUFFLE, 265
- EWL_ICON_MEDIA_RECORD, 265
- EWL_ICON_MEDIA_SEEK_-
BACKWARD, 265
- EWL_ICON_MEDIA_SEEK_-
FORWARD, 266
- EWL_ICON_MEDIA_SKIP_-
BACKWARD, 266
- EWL_ICON_MEDIA_SKIP_-
FORWARD, 266
- EWL_ICON_MULTIMEDIA_PLAYER,
266
- EWL_ICON_MULTIMEDIA_-
VOLUME_CONTROL, 266
- EWL_ICON_NETWORK_ERROR, 266
- EWL_ICON_NETWORK_IDLE, 266
- EWL_ICON_NETWORK_OFFLINE,
266
- EWL_ICON_NETWORK_RECEIVE,
266
- EWL_ICON_NETWORK_SERVER,
266
- EWL_ICON_NETWORK_TRANSMIT,
266
- EWL_ICON_NETWORK_-
TRANSMIT_RECEIVE, 267
- EWL_ICON_NETWORK_WIRED, 267
- EWL_ICON_NETWORK_WIRELESS,
267
- EWL_ICON_NETWORK_-
WORKGROUP, 267
- EWL_ICON_PACKAGE_X_-
GENERIC, 267
- EWL_ICON_PREFERENCES_-
DESKTOP, 267
- EWL_ICON_PREFERENCES_-
DESKTOP_ACCESSIBILITY,
267
- EWL_ICON_PREFERENCES_-
DESKTOP_FONT, 267
- EWL_ICON_PREFERENCES_-
DESKTOP_KEYBOARD, 267
- EWL_ICON_PREFERENCES_-
DESKTOP_LOCALE, 268
- EWL_ICON_PREFERENCES_-
DESKTOP_MULTIMEDIA, 268
- EWL_ICON_PREFERENCES_-
DESKTOP_PERIPHERALS, 268
- EWL_ICON_PREFERENCES_-
DESKTOP_PERSONAL, 268
- EWL_ICON_PREFERENCES_-
DESKTOP_SCREENSAVER, 268
- EWL_ICON_PREFERENCES_-
DESKTOP_THEME, 268
- EWL_ICON_PREFERENCES_-
DESKTOP_WALLPAPER, 268
- EWL_ICON_PREFERENCES_-
OTHER, 268
- EWL_ICON_PREFERENCES_-
SYSTEM, 268
- EWL_ICON_PREFERENCES_-
SYSTEM_NETWORK, 268
- EWL_ICON_PRINTER, 269
- EWL_ICON_PRINTER_ERROR, 269
- EWL_ICON_PRINTER_PRINTING,
269
- EWL_ICON_PROCESS_WORKING,
269
- EWL_ICON_SIZE_LARGE, 269
- EWL_ICON_SIZE_MEDIUM, 269
- EWL_ICON_SIZE_SMALL, 269
- EWL_ICON_SOFTWARE_UPDATE_-
AVAILABLE, 269
- EWL_ICON_SOFTWARE_UPDATE_-
URGENT, 269
- EWL_ICON_START_HERE, 269
- EWL_ICON_SYNC_ERROR, 269
- EWL_ICON_SYNC_-
SYNCHRONIZING, 270
- EWL_ICON_SYSTEM_FILE_-
MANAGER, 270
- EWL_ICON_SYSTEM_HELP, 270
- EWL_ICON_SYSTEM_LOCK_-
SCREEN, 270
- EWL_ICON_SYSTEM_LOG_OUT, 270
- EWL_ICON_SYSTEM_RUN, 270
- EWL_ICON_SYSTEM_SEARCH, 270
- EWL_ICON_SYSTEM_SOFTWARE_-
UPDATE, 270
- EWL_ICON_TASK_DUE, 270
- EWL_ICON_TASK_PASSED_DUE,
270
- EWL_ICON_TEXT_HTML, 270
- EWL_ICON_TEXT_X_GENERIC, 271
- EWL_ICON_TEXT_X_GENERIC_-
TEMPLATE, 271

- EWL_ICON_TEXT_X_SCRIPT, 271
- ewl_icon_theme_icon_path_get, 274
- ewl_icon_theme_init, 274
- ewl_icon_theme_shutdown, 275
- ewl_icon_theme_theme_change, 275
- EWL_ICON_TOOLS_CHECK_-
SPELLING, 271
- EWL_ICON_USER_AWAY, 271
- EWL_ICON_USER_DESKTOP, 271
- EWL_ICON_USER_HOME, 271
- EWL_ICON_USER_IDLE, 271
- EWL_ICON_USER_OFFLINE, 271
- EWL_ICON_USER_ONLINE, 271
- EWL_ICON_USER_TRASH, 271
- EWL_ICON_USER_TRASH_FULL,
272
- EWL_ICON_UTILITIES_TERMINAL,
272
- EWL_ICON_VIDEO_X_GENERIC,
272
- EWL_ICON_VIEW_FULLSCREEN,
272
- EWL_ICON_VIEW_REFRESH, 272
- EWL_ICON_VIEW_SORT_-
ASCENDING, 272
- EWL_ICON_VIEW_SORT_-
DESCENDING, 272
- EWL_ICON_WEATHER_CLEAR, 272
- EWL_ICON_WEATHER_CLEAR_-
NIGHT, 272
- EWL_ICON_WEATHER_FEW_-
CLOUDS, 272
- EWL_ICON_WEATHER_FEW_-
CLOUDS_NIGHT, 272
- EWL_ICON_WEATHER_FOG, 273
- EWL_ICON_WEATHER_OVERCAST,
273
- EWL_ICON_WEATHER_SEVERE_-
ALERT, 273
- EWL_ICON_WEATHER_SHOWERS,
273
- EWL_ICON_WEATHER_SHOWERS_-
SCATTERED, 273
- EWL_ICON_WEATHER_SNOW, 273
- EWL_ICON_WEATHER_STORM, 273
- EWL_ICON_WINDOW_CLOSE, 273
- EWL_ICON_WINDOW_NEW, 273
- EWL_ICON_X_OFFICE_ADDRESS_-
BOOK, 273
- EWL_ICON_X_OFFICE_CALENDAR,
273
- EWL_ICON_X_OFFICE_-
DOCUMENT, 274
- EWL_ICON_X_OFFICE_-
PRESENTATION, 274
- EWL_ICON_X_OFFICE_-
SPREADSHEET, 274
- EWL_ICON_ZOOM_BEST_FIT, 274
- EWL_ICON_ZOOM_IN, 274
- EWL_ICON_ZOOM_ORIGINAL, 274
- EWL_ICON_ZOOM_OUT, 274
- Ewl_Icon_Theme: The icon theme code, 245
- ewl_icon_theme_icon_path_get
Ewl_Icon_Theme, 274
- ewl_icon_theme_init
Ewl_Icon_Theme, 274
- EWL_ICON_THEME_PER_LOOP
bin/tests/icon_theme/ewl_icon_theme.c,
1049
- ewl_icon_theme_shutdown
Ewl_Icon_Theme, 275
- ewl_icon_theme_theme_change
Ewl_Icon_Theme, 275
- ewl_icon_thumbnailing_get
Ewl_Icon, 243
- ewl_icon_thumbnailing_set
Ewl_Icon, 244
- EWL_ICON_TOOLS_CHECK_SPELLING
Ewl_Icon_Theme, 271
- EWL_ICON_TYPE
Ewl_Icon, 238
- Ewl_Icon_Type
Ewl_Enums, 152, 158
- ewl_icon_type_get
Ewl_Icon, 244
- EWL_ICON_TYPE_LONG
Ewl_Enums, 158
- ewl_icon_type_set
Ewl_Icon, 244
- EWL_ICON_TYPE_SHORT
Ewl_Enums, 158
- EWL_ICON_USER_AWAY
Ewl_Icon_Theme, 271
- EWL_ICON_USER_DESKTOP
Ewl_Icon_Theme, 271
- EWL_ICON_USER_HOME
Ewl_Icon_Theme, 271
- EWL_ICON_USER_IDLE
Ewl_Icon_Theme, 271
- EWL_ICON_USER_OFFLINE
Ewl_Icon_Theme, 271
- EWL_ICON_USER_ONLINE
Ewl_Icon_Theme, 271
- EWL_ICON_USER_TRASH
Ewl_Icon_Theme, 271
- EWL_ICON_USER_TRASH_FULL
Ewl_Icon_Theme, 272

- EWL_ICON_UTILITIES_TERMINAL
 - Ewl_Icon_Theme, 272
- EWL_ICON_VIDEO_X_GENERIC
 - Ewl_Icon_Theme, 272
- EWL_ICON_VIEW_FULLSCREEN
 - Ewl_Icon_Theme, 272
- EWL_ICON_VIEW_REFRESH
 - Ewl_Icon_Theme, 272
- EWL_ICON_VIEW_SORT_ASCENDING
 - Ewl_Icon_Theme, 272
- EWL_ICON_VIEW_SORT_DESCENDING
 - Ewl_Icon_Theme, 272
- EWL_ICON_WEATHER_CLEAR
 - Ewl_Icon_Theme, 272
- EWL_ICON_WEATHER_CLEAR_NIGHT
 - Ewl_Icon_Theme, 272
- EWL_ICON_WEATHER_FEW_CLOUDS
 - Ewl_Icon_Theme, 272
- EWL_ICON_WEATHER_FEW_-
 - CLOUDS_NIGHT
 - Ewl_Icon_Theme, 272
- EWL_ICON_WEATHER_FOG
 - Ewl_Icon_Theme, 273
- EWL_ICON_WEATHER_OVERCAST
 - Ewl_Icon_Theme, 273
- EWL_ICON_WEATHER_SEVERE_-
 - ALERT
 - Ewl_Icon_Theme, 273
- EWL_ICON_WEATHER_SHOWERS
 - Ewl_Icon_Theme, 273
- EWL_ICON_WEATHER_SHOWERS_-
 - SCATTERED
 - Ewl_Icon_Theme, 273
- EWL_ICON_WEATHER_SNOW
 - Ewl_Icon_Theme, 273
- EWL_ICON_WEATHER_STORM
 - Ewl_Icon_Theme, 273
- EWL_ICON_WINDOW_CLOSE
 - Ewl_Icon_Theme, 273
- EWL_ICON_WINDOW_NEW
 - Ewl_Icon_Theme, 273
- EWL_ICON_X_OFFICE_ADDRESS_-
 - BOOK
 - Ewl_Icon_Theme, 273
- EWL_ICON_X_OFFICE_CALENDAR
 - Ewl_Icon_Theme, 273
- EWL_ICON_X_OFFICE_DOCUMENT
 - Ewl_Icon_Theme, 274
- EWL_ICON_X_OFFICE_PRESENTATION
 - Ewl_Icon_Theme, 274
- EWL_ICON_X_OFFICE_SPREADSHEET
 - Ewl_Icon_Theme, 274
- EWL_ICON_ZOOM_BEST_FIT
 - Ewl_Icon_Theme, 274
- EWL_ICON_ZOOM_IN
 - Ewl_Icon_Theme, 274
- EWL_ICON_ZOOM_ORIGINAL
 - Ewl_Icon_Theme, 274
- EWL_ICON_ZOOM_OUT
 - Ewl_Icon_Theme, 274
- EWL_IMAGE
 - Ewl_Image, 279
- Ewl_Image, 686
 - ah, 686
 - aw, 686
 - cs, 687
 - EWL_IMAGE, 279
 - Ewl_Image, 279
 - ewl_image_cb_configure, 280
 - ewl_image_cb_destroy, 280
 - ewl_image_cb_mouse_down, 280
 - ewl_image_cb_mouse_move, 280
 - ewl_image_cb_mouse_up, 280
 - ewl_image_cb_obscure, 280
 - ewl_image_cb_reveal, 280
 - ewl_image_constrain_get, 280
 - ewl_image_constrain_set, 280
 - ewl_image_file_key_get, 280
 - ewl_image_file_key_set, 281
 - ewl_image_file_path_get, 281
 - ewl_image_file_path_set, 281
 - ewl_image_file_set, 281
 - ewl_image_flip, 282
 - ewl_image_init, 282
 - EWL_IMAGE_IS, 279
 - ewl_image_new, 282
 - ewl_image_proportional_get, 282
 - ewl_image_proportional_set, 282
 - ewl_image_rotate, 283
 - ewl_image_scale_get, 283
 - ewl_image_scale_set, 283
 - ewl_image_size_get, 284
 - ewl_image_size_set, 284
 - EWL_IMAGE_THUMBNAIL, 279
 - Ewl_Image_Thumbnail, 279
 - ewl_image_thumbnail_get, 284
 - ewl_image_thumbnail_init, 284
 - ewl_image_thumbnail_new, 285
 - ewl_image_thumbnail_request, 285
 - EWL_IMAGE_THUMBNAIL_TYPE, 279
 - ewl_image_tile_set, 285
 - EWL_IMAGE_TYPE, 279
 - ewl_image_view_get, 285
 - h, 687
 - image, 687
 - key, 687
 - oh, 687

- ow, 687
- path, 687
- proportional, 687
- set, 687
- sh, 687
- sw, 687
- tile, 687
- type, 688
- w, 688
- widget, 688
- x, 688
- y, 688
- Ewl_Image: An Image Display Widget, 276
- ewl_image_cb_configure
 - Ewl_Image, 280
 - lib/ewl_image.c, 1059
- ewl_image_cb_destroy
 - Ewl_Image, 280
 - lib/ewl_image.c, 1059
- ewl_image_cb_mouse_down
 - Ewl_Image, 280
 - lib/ewl_image.c, 1059
- ewl_image_cb_mouse_move
 - Ewl_Image, 280
 - lib/ewl_image.c, 1059
- ewl_image_cb_mouse_up
 - Ewl_Image, 280
 - lib/ewl_image.c, 1059
- ewl_image_cb_obscure
 - Ewl_Image, 280
 - lib/ewl_image.c, 1059
- ewl_image_cb_reparent
 - lib/ewl_image.c, 1059
- ewl_image_cb_reveal
 - Ewl_Image, 280
 - lib/ewl_image.c, 1059
- ewl_image_constrain_get
 - Ewl_Image, 280
- ewl_image_constrain_set
 - Ewl_Image, 280
- ewl_image_file_key_get
 - Ewl_Image, 280
- ewl_image_file_key_set
 - Ewl_Image, 281
- ewl_image_file_path_get
 - Ewl_Image, 281
- ewl_image_file_path_set
 - Ewl_Image, 281
- ewl_image_file_set
 - Ewl_Image, 281
- ewl_image_flip
 - Ewl_Image, 282
- ewl_image_init
 - Ewl_Image, 282
- EWL_IMAGE_IS
 - Ewl_Image, 279
- ewl_image_new
 - Ewl_Image, 282
- ewl_image_proportional_get
 - Ewl_Image, 282
- ewl_image_proportional_set
 - Ewl_Image, 282
- ewl_image_rotate
 - Ewl_Image, 283
- ewl_image_scale_get
 - Ewl_Image, 283
- ewl_image_scale_set
 - Ewl_Image, 283
- ewl_image_size_get
 - Ewl_Image, 284
- ewl_image_size_set
 - Ewl_Image, 284
- EWL_IMAGE_THUMBNAIL
 - Ewl_Image, 279
- Ewl_Image_Thumbnail, 689
 - Ewl_Image, 279
 - image, 689
 - orig, 689
 - thumb, 689
- ewl_image_thumbnail.c
 - test_info, 810
- ewl_image_thumbnail_get
 - Ewl_Image, 284
- ewl_image_thumbnail_init
 - Ewl_Image, 284
- ewl_image_thumbnail_new
 - Ewl_Image, 285
- ewl_image_thumbnail_request
 - Ewl_Image, 285
- EWL_IMAGE_THUMBNAIL_TYPE
 - Ewl_Image, 279
- ewl_image_tile_set
 - Ewl_Image, 285
- EWL_IMAGE_TYPE
 - Ewl_Image, 279
- Ewl_Image_Type
 - Ewl_Enums, 152, 158
- EWL_IMAGE_TYPE_EDJE
 - Ewl_Enums, 158
- EWL_IMAGE_TYPE_NORMAL
 - Ewl_Enums, 158
- ewl_image_view_get
 - Ewl_Image, 285
- ewl_init
 - Ewl_Misc, 313
- ewl_io_manager.h
 - ewl_io_manager_extension_icon_-
name_get, 1070

- ewl_io_manager_init, 1070
- ewl_io_manager_mime_type_icon_name_get, 1070
- Ewl_IO_Manager_Plugin, 1070
- ewl_io_manager_shutdown, 1070
- ewl_io_manager_string_read, 1071
- ewl_io_manager_string_write, 1071
- ewl_io_manager_uri_mime_type_get, 1071
- ewl_io_manager_uri_read, 1071
- ewl_io_manager_uri_write, 1072
- ewl_io_manager_extension_icon_name_get
- ewl_io_manager.h, 1070
- lib/ewl_io_manager.c, 1065
- ewl_io_manager_init
- ewl_io_manager.h, 1070
- lib/ewl_io_manager.c, 1065
- ewl_io_manager_mime_type_icon_name_get
- ewl_io_manager.h, 1070
- lib/ewl_io_manager.c, 1065
- Ewl_IO_Manager_Plugin, 690
- ewl_io_manager.h, 1070
- handle, 690
- string_read, 690
- string_write, 690
- uri_read, 690
- uri_write, 690
- ewl_io_manager_shutdown
- ewl_io_manager.h, 1070
- lib/ewl_io_manager.c, 1065
- ewl_io_manager_string_read
- ewl_io_manager.h, 1071
- lib/ewl_io_manager.c, 1065
- ewl_io_manager_string_write
- ewl_io_manager.h, 1071
- lib/ewl_io_manager.c, 1066
- ewl_io_manager_uri_mime_type_get
- ewl_io_manager.h, 1071
- lib/ewl_io_manager.c, 1066
- ewl_io_manager_uri_read
- ewl_io_manager.h, 1071
- lib/ewl_io_manager.c, 1066
- ewl_io_manager_uri_write
- ewl_io_manager.h, 1072
- lib/ewl_io_manager.c, 1066
- EWL_KEY_MODIFIER_ALT
- Ewl_Enums, 158
- EWL_KEY_MODIFIER_CTRL
- Ewl_Enums, 158
- EWL_KEY_MODIFIER_MOD
- Ewl_Enums, 158
- EWL_KEY_MODIFIER_SHIFT
- Ewl_Enums, 158
- EWL_KEY_MODIFIER_WIN
- Ewl_Enums, 158
- Ewl_Key_Modifiers
- Ewl_Enums, 152, 158
- EWL_LABEL
- Ewl_Label, 287
- Ewl_Label, 692
- EWL_LABEL, 287
- Ewl_Label, 288
- ewl_label_init, 288
- EWL_LABEL_IS, 287
- ewl_label_new, 288
- ewl_label_text_get, 288
- ewl_label_text_set, 288
- EWL_LABEL_TYPE, 288
- ewl_label_view_get, 289
- widget, 692
- Ewl_Label: A fully-themable text label, 287
- ewl_label_init
- Ewl_Label, 288
- EWL_LABEL_IS
- Ewl_Label, 287
- ewl_label_new
- Ewl_Label, 288
- ewl_label_text_get
- Ewl_Label, 288
- ewl_label_text_set
- Ewl_Label, 288
- EWL_LABEL_TYPE
- Ewl_Label, 288
- ewl_label_view_get
- Ewl_Label, 289
- ewl_layer.c
- test_info, 811
- EWL_LIST
- Ewl_List, 290
- Ewl_List, 693
- EWL_LIST, 290
- Ewl_List, 291
- ewl_list_cb_child_add, 291
- ewl_list_cb_configure, 291
- ewl_list_cb_item_clicked, 291
- ewl_list_cb_selected_change, 291
- ewl_list_init, 291
- EWL_LIST_IS, 290
- ewl_list_new, 291
- EWL_LIST_TYPE, 290
- mvc, 693
- Ewl_List: A list widget, 290
- ewl_list_cb_child_add
- Ewl_List, 291
- ewl_list_cb_configure
- Ewl_List, 291
- lib/ewl_list.c, 1076

- ewl_list_cb_item_clicked
 - Ewl_List, 291
 - lib/ewl_list.c, 1076
- ewl_list_cb_selected_change
 - Ewl_List, 291
- ewl_list_init
 - Ewl_List, 291
- EWL_LIST_IS
 - Ewl_List, 290
- ewl_list_new
 - Ewl_List, 291
- EWL_LIST_TYPE
 - Ewl_List, 290
- Ewl_Macros
 - FREE, 292
 - IF_FREE, 292
 - IF_FREE_HASH, 292
 - IF_FREE_LIST, 293
 - IF_RELEASE, 293
 - MAX, 293
 - MIN, 293
 - NEW, 293
 - REALLOC, 293
 - ZERO, 293
- Ewl_Macros: Useful Macros Used Internally and Available Externally, 292
- ewl_main
 - Ewl_Misc, 313
- ewl_main_quit
 - Ewl_Misc, 314
- EWL_MEDIA
 - Ewl_Media, 296
- Ewl_Media, 694
 - block_seek, 694
 - EWL_MEDIA, 296
 - Ewl_Media, 297
 - ewl_media_audio_mute_get, 297
 - ewl_media_audio_mute_set, 297
 - ewl_media_audio_volume_get, 297
 - ewl_media_audio_volume_set, 297
 - ewl_media_cb_configure, 298
 - ewl_media_cb_realize, 298
 - ewl_media_cb_reveal, 298
 - ewl_media_cb_unrealize, 298
 - ewl_media_init, 298
 - EWL_MEDIA_IS, 296
 - ewl_media_is_available, 298
 - ewl_media_length_get, 298
 - ewl_media_length_time_get, 299
 - ewl_media_media_get, 299
 - ewl_media_media_set, 299
 - ewl_media_module_get, 299
 - ewl_media_module_set, 300
 - ewl_media_new, 300
 - ewl_media_play_set, 300
 - ewl_media_position_get, 300
 - ewl_media_position_set, 301
 - ewl_media_position_time_get, 301
 - ewl_media_seekable_get, 301
 - EWL_MEDIA_TYPE, 297
 - media, 694
 - module, 694
 - mute, 694
 - position, 695
 - video, 695
 - volume, 695
 - widget, 695
- Ewl_Media: A Multi media widget, 295
- ewl_media_audio_mute_get
 - Ewl_Media, 297
- ewl_media_audio_mute_set
 - Ewl_Media, 297
- ewl_media_audio_volume_get
 - Ewl_Media, 297
- ewl_media_audio_volume_set
 - Ewl_Media, 297
- ewl_media_cb_configure
 - Ewl_Media, 298
 - lib/ewl_media.c, 1083
- ewl_media_cb_realize
 - Ewl_Media, 298
 - lib/ewl_media.c, 1083
- ewl_media_cb_reveal
 - Ewl_Media, 298
 - lib/ewl_media.c, 1083
- ewl_media_cb_unrealize
 - Ewl_Media, 298
 - lib/ewl_media.c, 1083
- ewl_media_init
 - Ewl_Media, 298
- EWL_MEDIA_IS
 - Ewl_Media, 296
- ewl_media_is_available
 - Ewl_Media, 298
- ewl_media_length_get
 - Ewl_Media, 298
- ewl_media_length_time_get
 - Ewl_Media, 299
- ewl_media_media_get
 - Ewl_Media, 299
- ewl_media_media_set
 - Ewl_Media, 299
- ewl_media_module_get
 - Ewl_Media, 299
- EWL_MEDIA_MODULE_GSTREAMER
 - Ewl_Enums, 158
- ewl_media_module_set
 - Ewl_Media, 300

- Ewl_Media_Module_Type
 - Ewl_Enums, 152, 158
- EWL_MEDIA_MODULE_XINE
 - Ewl_Enums, 158
- ewl_media_new
 - Ewl_Media, 300
- ewl_media_play_set
 - Ewl_Media, 300
- ewl_media_position_get
 - Ewl_Media, 300
- ewl_media_position_set
 - Ewl_Media, 301
- ewl_media_position_time_get
 - Ewl_Media, 301
- ewl_media_seekable_get
 - Ewl_Media, 301
- EWL_MEDIA_TYPE
 - Ewl_Media, 297
- EWL_MENU
 - Ewl_Menu, 303
- Ewl_Menu, 696
 - EWL_MENU, 303
 - Ewl_Menu, 303
 - ewl_menu_cb_configure, 304
 - ewl_menu_cb_destroy, 304
 - ewl_menu_cb_expand, 304
 - ewl_menu_cb_hide, 304
 - ewl_menu_cb_mouse_move, 304
 - ewl_menu_cb_popup_destroy, 304
 - ewl_menu_cb_realize, 304
 - ewl_menu_collapse, 304
 - ewl_menu_from_info, 304
 - Ewl_Menu_Info, 303
 - ewl_menu_init, 304
 - EWL_MENU_IS, 303
 - ewl_menu_mouse_feed, 305
 - ewl_menu_new, 305
 - EWL_MENU_TYPE, 303
 - item, 696
 - menubar_parent, 696
 - popup, 697
- Ewl_Menu: A Simple Windowed Menu, 302
- ewl_menu_cb_configure
 - Ewl_Menu, 304
 - lib/ewl_menu.c, 1088
- ewl_menu_cb_destroy
 - Ewl_Menu, 304
 - lib/ewl_menu.c, 1088
- ewl_menu_cb_expand
 - Ewl_Menu, 304
 - lib/ewl_menu.c, 1088
- ewl_menu_cb_hide
 - Ewl_Menu, 304
- ewl_menu_cb_mouse_move
 - Ewl_Menu, 304
- lib/ewl_menu.c, 1088
- ewl_menu_cb_popup_destroy
 - Ewl_Menu, 304
 - lib/ewl_menu.c, 1088
- ewl_menu_cb_realize
 - Ewl_Menu, 304
- ewl_menu_collapse
 - Ewl_Menu, 304
- ewl_menu_from_info
 - Ewl_Menu, 304
- Ewl_Menu_Info, 698
 - cb, 698
 - Ewl_Menu, 303
 - img, 698
 - name, 698
- ewl_menu_init
 - Ewl_Menu, 304
- EWL_MENU_IS
 - Ewl_Menu, 303
- EWL_MENU_ITEM
 - Ewl_Menu_Item, 306
- Ewl_Menu_Item, 699
 - button, 699
 - EWL_MENU_ITEM, 306
 - Ewl_Menu_Item, 307
 - ewl_menu_item_init, 307
 - EWL_MENU_ITEM_IS, 306
 - ewl_menu_item_new, 307
 - EWL_MENU_ITEM_TYPE, 306
 - inmenu, 699
- Ewl_Menu_Item: The basic menu item, 306
- ewl_menu_item_init
 - Ewl_Menu_Item, 307
- EWL_MENU_ITEM_IS
 - Ewl_Menu_Item, 306
- ewl_menu_item_new
 - Ewl_Menu_Item, 307
- EWL_MENU_ITEM_TYPE
 - Ewl_Menu_Item, 306
- ewl_menu_mouse_feed
 - Ewl_Menu, 305
- ewl_menu_new
 - Ewl_Menu, 305
- EWL_MENU_TYPE
 - Ewl_Menu, 303
- EWL_MENUBAR
 - Ewl_Menubar, 309
- Ewl_Menubar, 701
 - ewl_hmenubar_new, 309
 - EWL_MENUBAR, 309
 - Ewl_Menubar, 309
 - ewl_menubar_cb_child_add, 309
 - ewl_menubar_from_info, 309

- Ewl_Menubar_Info, 309
- ewl_menubar_init, 310
- EWL_MENUBAR_IS, 309
- ewl_menubar_new, 310
- ewl_menubar_orientation_get, 310
- ewl_menubar_orientation_set, 310
- EWL_MENUBAR_TYPE, 309
- ewl_vmenubar_new, 310
- inner_box, 701
- outer_box, 701
- Ewl_Menubar: A menu bar widget, 308
- ewl_menubar_cb_child_add
 - Ewl_Menubar, 309
- ewl_menubar_from_info
 - Ewl_Menubar, 309
- Ewl_Menubar_Info, 702
 - Ewl_Menubar, 309
 - menu, 702
 - name, 702
- ewl_menubar_init
 - Ewl_Menubar, 310
- EWL_MENUBAR_IS
 - Ewl_Menubar, 309
- ewl_menubar_new
 - Ewl_Menubar, 310
- ewl_menubar_orientation_get
 - Ewl_Menubar, 310
- ewl_menubar_orientation_set
 - Ewl_Menubar, 310
- EWL_MENUBAR_TYPE
 - Ewl_Menubar, 309
- Ewl_Misc
 - ewl_canvas_destroy, 313
 - ewl_canvas_object_destroy, 313
 - ewl_configure_request, 313
 - ewl_debug_indent_print, 313
 - ewl_destroy_request, 313
 - ewl_init, 313
 - ewl_main, 313
 - ewl_main_quit, 314
 - ewl_print_help, 314
 - ewl_realize_cancel_request, 314
 - ewl_realize_request, 314
 - ewl_shutdown, 314
 - ewl_shutdown_add, 314
 - Ewl_Shutdown_Hook, 313
- ewl_misc.c
 - configure_active, 1101
 - configure_available, 1101
 - ewl_backtrace, 1101
 - Ewl_Configure_Queue, 1101
 - EWL_CONFIGURE_QUEUE_SIZE, 1101
 - ewl_embed_list, 1101
 - EWL_GC_LIMIT, 1101
 - ewl_print_warning, 1101
 - ewl_segv, 1101
 - ewl_window_list, 1101
- Ewl_Misc: Miscellaneous Utility Functions, 312
- ewl_modal.c
 - test_info, 812
- EWL_MODEL
 - Ewl_Model, 317
- Ewl_Model, 703
 - count, 703
 - data, 703
 - data_free, 703
 - EWL_MODEL, 317
 - Ewl_Model, 318
 - ewl_model_cb_ecore_list_count, 319
 - ewl_model_cb_ecore_list_fetch, 320
 - EWL_MODEL_COLUMN_SORTABLE, 317
 - Ewl_Model_Column_Sortable, 318
 - ewl_model_column_sortable_get, 320
 - ewl_model_column_sortable_set, 320
 - EWL_MODEL_DATA_COUNT, 317
 - Ewl_Model_Data_Count, 318
 - ewl_model_data_count_get, 320
 - ewl_model_data_count_set, 321
 - EWL_MODEL_DATA_EXPANDABLE, 318
 - Ewl_Model_Data_Expandable, 319
 - ewl_model_data_expandable_get, 321
 - ewl_model_data_expandable_set, 321
 - EWL_MODEL_DATA_FETCH, 318
 - Ewl_Model_Data_Fetch, 319
 - ewl_model_data_fetch_get, 321
 - ewl_model_data_fetch_set, 322
 - Ewl_Model_Data_Free, 319
 - ewl_model_data_free_get, 322
 - ewl_model_data_free_set, 322
 - EWL_MODEL_DATA_HEADER_FETCH, 318
 - Ewl_Model_Data_Header_Fetch, 319
 - ewl_model_data_header_fetch_get, 322
 - ewl_model_data_header_fetch_set, 323
 - EWL_MODEL_DATA_SORT, 318
 - Ewl_Model_Data_Sort, 319
 - ewl_model_data_sort_get, 323
 - ewl_model_data_sort_set, 323
 - ewl_model_ecore_list_get, 323
 - EWL_MODEL_EXPANSION_DATA_FETCH, 318
 - Ewl_Model_Expansion_Data_Fetch, 319
 - ewl_model_expansion_data_fetch_get, 324

- ewl_model_expansion_data_fetch_set, 324
- EWL_MODEL_EXPANSION_MODEL_FETCH, 318
- Ewl_Model_Expansion_Model_Fetch, 319
- ewl_model_expansion_model_fetch_get, 324
- ewl_model_expansion_model_fetch_set, 324
- EWL_MODEL_FREE, 318
- ewl_model_init, 325
- ewl_model_new, 325
- expansion, 703
- fetch, 703
- header, 703
- is, 703
- model, 704
- sort, 704
- sortable, 704
- Ewl_Model: A data model, 315
- ewl_model_cb_ecore_list_count Ewl_Model, 319
- ewl_model_cb_ecore_list_fetch Ewl_Model, 320
- EWL_MODEL_COLUMN_SORTABLE Ewl_Model, 317
- Ewl_Model_Column_Sortable Ewl_Model, 318
- ewl_model_column_sortable_get Ewl_Model, 320
- ewl_model_column_sortable_set Ewl_Model, 320
- EWL_MODEL_DATA_COUNT Ewl_Model, 317
- Ewl_Model_Data_Count Ewl_Model, 318
- ewl_model_data_count_get Ewl_Model, 320
- ewl_model_data_count_set Ewl_Model, 321
- EWL_MODEL_DATA_EXPANDABLE Ewl_Model, 318
- Ewl_Model_Data_Expandable Ewl_Model, 319
- ewl_model_data_expandable_get Ewl_Model, 321
- ewl_model_data_expandable_set Ewl_Model, 321
- EWL_MODEL_DATA_FETCH Ewl_Model, 318
- Ewl_Model_Data_Fetch Ewl_Model, 319
- ewl_model_data_fetch_get Ewl_Model, 321
- ewl_model_data_fetch_set Ewl_Model, 322
- EWL_MODEL_DATA_HEADER_FETCH Ewl_Model, 318
- Ewl_Model_Data_Header_Fetch Ewl_Model, 319
- ewl_model_data_header_fetch_get Ewl_Model, 322
- ewl_model_data_header_fetch_set Ewl_Model, 323
- EWL_MODEL_DATA_SORT Ewl_Model, 318
- Ewl_Model_Data_Sort Ewl_Model, 319
- ewl_model_data_sort_get Ewl_Model, 323
- ewl_model_data_sort_set Ewl_Model, 323
- ewl_model_ecore_list_get Ewl_Model, 323
- EWL_MODEL_EXPANSION_DATA_FETCH Ewl_Model, 318
- Ewl_Model_Expansion_Data_Fetch Ewl_Model, 319
- ewl_model_expansion_data_fetch_get Ewl_Model, 324
- ewl_model_expansion_data_fetch_set Ewl_Model, 324
- EWL_MODEL_EXPANSION_MODEL_FETCH Ewl_Model, 318
- Ewl_Model_Expansion_Model_Fetch Ewl_Model, 319
- ewl_model_expansion_model_fetch_get Ewl_Model, 324
- ewl_model_expansion_model_fetch_set Ewl_Model, 324
- EWL_MODEL_FREE Ewl_Model, 318
- ewl_model_init Ewl_Model, 325
- ewl_model_new Ewl_Model, 325
- EWL_MOUSE_CURSOR_ARROW Ewl_Enums, 159

- EWL_MOUSE_CURSOR_BASED_-
 ARROW_DOWN
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_BOAT
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_BOTTOM_-
 LEFT_CORNER
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_BOTTOM_-
 RIGHT_CORNER
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_BOTTOM_SIDE
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_BOTTOM_TEE
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_BOX_SPIRAL
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_CENTER_PTR
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_CIRCLE
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_CLOCK
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_COFFEE_MUG
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_CROSS
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_CROSS_-
 REVERSE
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_CROSSHAIR
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_DIAMOND_-
 CROSS
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_DOT
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_DOT_BOX_-
 MASK
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_DOUBLE_-
 ARROW
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_DRAFT_LARGE
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_DRAFT_SMALL
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_DRAPED_BOX
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_EXCHANGE
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_FLEUR
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_GOBBLER
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_GUMBY
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_HAND1
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_HAND2
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_HEART
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_ICON
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_IRON_CROSS
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_LEFT_PTR
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_LEFT_SIDE
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_LEFT_TEE
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_LEFTBUTTON
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_LL_ANGLE
 Ewl_Enums, 159
- EWL_MOUSE_CURSOR_LR_ANGLE
 Ewl_Enums, 160
- EWL_MOUSE_CURSOR_MAN
 Ewl_Enums, 160
- EWL_MOUSE_CURSOR_MAX
 Ewl_Enums, 160
- EWL_MOUSE_CURSOR_-
 MIDDLEBUTTON
 Ewl_Enums, 160
- EWL_MOUSE_CURSOR_MOUSE
 Ewl_Enums, 160
- EWL_MOUSE_CURSOR_PENCIL
 Ewl_Enums, 160
- EWL_MOUSE_CURSOR_PIRATE
 Ewl_Enums, 160
- EWL_MOUSE_CURSOR_PLUS
 Ewl_Enums, 160
- EWL_MOUSE_CURSOR_QUESTION_-
 ARROW
 Ewl_Enums, 160
- EWL_MOUSE_CURSOR_RIGHT_PTR
 Ewl_Enums, 160
- EWL_MOUSE_CURSOR_RIGHT_SIDE
 Ewl_Enums, 160
- EWL_MOUSE_CURSOR_RIGHT_TEE
 Ewl_Enums, 160
- EWL_MOUSE_CURSOR_RIGHTBUTTON
 Ewl_Enums, 160
- EWL_MOUSE_CURSOR_RTL_LOGO
 Ewl_Enums, 160
- EWL_MOUSE_CURSOR_SAILBOAT

- Ewl_Enums, 160
- EWL_MOUSE_CURSOR_SB_DOWN_-
ARROW
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_SB_H_-
DOUBLE_ARROW
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_SB_LEFT_-
ARROW
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_SB_RIGHT_-
ARROW
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_SB_UP_-
ARROW
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_SB_V_-
DOUBLE_ARROW
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_SHUTTLE
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_SIZING
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_SPIDER
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_SPRAYCAN
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_STAR
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_TARGET
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_TCROSS
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_TOP_LEFT_-
ARROW
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_TOP_LEFT_-
CORNER
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_TOP_RIGHT_-
CORNER
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_TOP_SIDE
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_TOP_TEE
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_TREK
 - Ewl_Enums, 160
- Ewl_Mouse_Cursor_Type
 - Ewl_Enums, 152, 158
- EWL_MOUSE_CURSOR_UL_ANGLE
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_UMBRELLA
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_UP
 - Ewl_Enums, 159
- EWL_MOUSE_CURSOR_UR_ANGLE
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_WATCH
 - Ewl_Enums, 160
- EWL_MOUSE_CURSOR_X
 - Ewl_Enums, 159
- EWL_MOUSE_CURSOR_XTERM
 - Ewl_Enums, 160
- EWL_MVC
 - Ewl_MVC, 328
- Ewl_MVC, 705
 - box, 705
 - cb, 705
 - data, 706
 - dirty, 706
 - EWL_MVC, 328
 - Ewl_MVC, 329
 - ewl_mvc_cb_destroy, 329
 - ewl_mvc_data_get, 329
 - ewl_mvc_data_set, 329
 - ewl_mvc_dirty_get, 330
 - ewl_mvc_dirty_set, 330
 - ewl_mvc_handle_click, 330
 - ewl_mvc_highlight, 330
 - ewl_mvc_init, 330
 - EWL_MVC_IS, 328
 - ewl_mvc_model_get, 330
 - ewl_mvc_model_set, 331
 - ewl_mvc_selected_add, 331
 - ewl_mvc_selected_change_cb_set, 331
 - ewl_mvc_selected_clear, 331
 - ewl_mvc_selected_count_get, 331
 - ewl_mvc_selected_get, 332
 - ewl_mvc_selected_is, 332
 - ewl_mvc_selected_list_get, 332
 - ewl_mvc_selected_list_set, 332
 - ewl_mvc_selected_range_add, 332
 - ewl_mvc_selected_rm, 333
 - ewl_mvc_selected_set, 333
 - ewl_mvc_selection_index_new, 333
 - ewl_mvc_selection_mode_get, 334
 - ewl_mvc_selection_mode_set, 334
 - ewl_mvc_selection_range_new, 334
 - EWL_MVC_TYPE, 328
 - ewl_mvc_view_change_cb_set, 334
 - ewl_mvc_view_get, 335
 - ewl_mvc_view_set, 335
 - EWL_SELECTION, 328
 - Ewl_Selection, 329
 - EWL_SELECTION_IDX, 328
 - Ewl_Selection_Idx, 329
 - EWL_SELECTION_RANGE, 329

- Ewl_Selection_Range, 329
- model, 706
- selected, 706
- selected_change, 706
- selection_mode, 706
- view, 706
- view_change, 706
- Ewl_MVC: The base model-view-controller framework, 326
- ewl_mvc_cb_destroy
 - Ewl_MVC, 329
 - lib/ewl_mvc.c, 1111
- ewl_mvc_data_get
 - Ewl_MVC, 329
- ewl_mvc_data_set
 - Ewl_MVC, 329
- ewl_mvc_dirty_get
 - Ewl_MVC, 330
- ewl_mvc_dirty_set
 - Ewl_MVC, 330
- ewl_mvc_handle_click
 - Ewl_MVC, 330
- ewl_mvc_highlight
 - Ewl_MVC, 330
- ewl_mvc_init
 - Ewl_MVC, 330
- EWL_MVC_IS
 - Ewl_MVC, 328
- ewl_mvc_model_get
 - Ewl_MVC, 330
- ewl_mvc_model_set
 - Ewl_MVC, 331
- ewl_mvc_selected_add
 - Ewl_MVC, 331
- ewl_mvc_selected_change_cb_set
 - Ewl_MVC, 331
- ewl_mvc_selected_clear
 - Ewl_MVC, 331
- ewl_mvc_selected_count_get
 - Ewl_MVC, 331
- ewl_mvc_selected_get
 - Ewl_MVC, 332
- ewl_mvc_selected_is
 - Ewl_MVC, 332
 - lib/ewl_mvc.c, 1111
- ewl_mvc_selected_list_get
 - Ewl_MVC, 332
- ewl_mvc_selected_list_set
 - Ewl_MVC, 332
- ewl_mvc_selected_range_add
 - Ewl_MVC, 332
- ewl_mvc_selected_rm
 - Ewl_MVC, 333
 - lib/ewl_mvc.c, 1111
- ewl_mvc_selected_set
 - Ewl_MVC, 333
- ewl_mvc_selection_index_new
 - Ewl_MVC, 333
- ewl_mvc_selection_mode_get
 - Ewl_MVC, 334
- ewl_mvc_selection_mode_set
 - Ewl_MVC, 334
- ewl_mvc_selection_range_new
 - Ewl_MVC, 334
- EWL_MVC_TYPE
 - Ewl_MVC, 328
- ewl_mvc_view_change_cb_set
 - Ewl_MVC, 334
- ewl_mvc_view_get
 - Ewl_MVC, 335
- ewl_mvc_view_set
 - Ewl_MVC, 335
- EWL_NOTEBOOK
 - Ewl_Notebook, 337
- Ewl_Notebook, 707
 - body, 707
 - box, 707
 - cur_page, 707
 - EWL_NOTEBOOK, 337
 - Ewl_Notebook, 338
 - ewl_notebook_cb_child_add, 338
 - ewl_notebook_cb_child_hide, 338
 - ewl_notebook_cb_child_remove, 338
 - ewl_notebook_cb_child_show, 338
 - ewl_notebook_cb_tab_clicked, 338
 - ewl_notebook_init, 338
 - EWL_NOTEBOOK_IS, 337
 - ewl_notebook_new, 338
 - ewl_notebook_page_tab_text_get, 338
 - ewl_notebook_page_tab_text_set, 339
 - ewl_notebook_page_tab_widget_get, 339
 - ewl_notebook_page_tab_widget_set, 339
 - ewl_notebook_tabbar_alignment_get, 339
 - ewl_notebook_tabbar_alignment_set, 340
 - ewl_notebook_tabbar_position_get, 340
 - ewl_notebook_tabbar_position_set, 340
 - ewl_notebook_tabbar_visible_get, 340
 - ewl_notebook_tabbar_visible_set, 341
 - EWL_NOTEBOOK_TYPE, 337
 - ewl_notebook_visible_page_get, 341
 - ewl_notebook_visible_page_set, 341
 - pages, 708
 - tabbar, 708
 - tabbar_position, 708

- Ewl_Notebook: A notebook widget, 336
- ewl_notebook_cb_child_add
 - Ewl_Notebook, 338
- ewl_notebook_cb_child_hide
 - Ewl_Notebook, 338
- ewl_notebook_cb_child_remove
 - Ewl_Notebook, 338
- ewl_notebook_cb_child_show
 - Ewl_Notebook, 338
- ewl_notebook_cb_tab_clicked
 - Ewl_Notebook, 338
- lib/ewl_notebook.c, 1118
- EWL_NOTEBOOK_FLAG_TABS_ -
 - HIDDEN
 - Ewl_Enums, 161
- Ewl_Notebook_Flags
 - Ewl_Enums, 152, 160
- ewl_notebook_init
 - Ewl_Notebook, 338
- EWL_NOTEBOOK_IS
 - Ewl_Notebook, 337
- ewl_notebook_new
 - Ewl_Notebook, 338
- ewl_notebook_page_tab_text_get
 - Ewl_Notebook, 338
- ewl_notebook_page_tab_text_set
 - Ewl_Notebook, 339
- ewl_notebook_page_tab_widget_get
 - Ewl_Notebook, 339
- ewl_notebook_page_tab_widget_set
 - Ewl_Notebook, 339
- ewl_notebook_tabbar_alignment_get
 - Ewl_Notebook, 339
- ewl_notebook_tabbar_alignment_set
 - Ewl_Notebook, 340
- ewl_notebook_tabbar_position_get
 - Ewl_Notebook, 340
- ewl_notebook_tabbar_position_set
 - Ewl_Notebook, 340
- ewl_notebook_tabbar_visible_get
 - Ewl_Notebook, 340
- ewl_notebook_tabbar_visible_set
 - Ewl_Notebook, 341
- EWL_NOTEBOOK_TYPE
 - Ewl_Notebook, 337
- ewl_notebook_visible_page_get
 - Ewl_Notebook, 341
- ewl_notebook_visible_page_set
 - Ewl_Notebook, 341
- EWL_OBJECT
 - Ewl_Object, 348
- Ewl_Object, 709
 - b, 710
 - CONFIGURED, 347
 - current, 710
 - CURRENT_H, 347
 - CURRENT_W, 348
 - CURRENT_X, 348
 - CURRENT_Y, 348
 - DESTROYED, 348
 - DISABLED, 348
 - EWL_OBJECT, 348
 - Ewl_Object, 356
 - ewl_object_alignment_get, 356
 - ewl_object_alignment_set, 356
 - ewl_object_current_geometry_get, 356
 - ewl_object_current_h_get, 357
 - ewl_object_current_size_get, 357
 - ewl_object_current_w_get, 357
 - ewl_object_current_x_get, 357
 - ewl_object_current_y_get, 358
 - ewl_object_custom_h_set, 348
 - ewl_object_custom_size_set, 348
 - ewl_object_custom_w_set, 349
 - ewl_object_fill_policy_get, 358
 - ewl_object_fill_policy_set, 358
 - ewl_object_flags_add, 358
 - ewl_object_flags_get, 349
 - ewl_object_flags_has, 349
 - ewl_object_flags_has_all, 350
 - ewl_object_flags_remove, 359
 - ewl_object_geometry_request, 359
 - ewl_object_h_request, 359
 - ewl_object_in_tab_list_get, 350
 - ewl_object_in_tab_list_set, 350
 - ewl_object_init, 360
 - ewl_object_insets_bottom_get, 360
 - ewl_object_insets_get, 360
 - ewl_object_insets_left_get, 360
 - ewl_object_insets_right_get, 361
 - ewl_object_insets_set, 361
 - ewl_object_insets_top_get, 361
 - EWL_OBJECT_MAX_SIZE, 351
 - ewl_object_maximum_h_get, 361
 - ewl_object_maximum_h_set, 362
 - ewl_object_maximum_size_get, 362
 - ewl_object_maximum_size_set, 362
 - ewl_object_maximum_w_get, 363
 - ewl_object_maximum_w_set, 363
 - EWL_OBJECT_MIN_SIZE, 351
 - ewl_object_minimum_h_get, 363
 - ewl_object_minimum_h_set, 363
 - ewl_object_minimum_size_get, 364
 - ewl_object_minimum_size_set, 364
 - ewl_object_minimum_w_get, 364
 - ewl_object_minimum_w_set, 365
 - ewl_object_padding_bottom_get, 365
 - ewl_object_padding_get, 365

- ewl_object_padding_left_get, 366
- ewl_object_padding_right_get, 366
- ewl_object_padding_set, 366
- ewl_object_padding_top_get, 366
- ewl_object_place, 367
- ewl_object_position_request, 367
- ewl_object_preferred_h_get, 367
- ewl_object_preferred_inner_h_get, 368
- ewl_object_preferred_inner_h_set, 368
- ewl_object_preferred_inner_size_get, 368
- ewl_object_preferred_inner_size_set, 368
- ewl_object_preferred_inner_w_get, 369
- ewl_object_preferred_inner_w_set, 369
- ewl_object_preferred_size_get, 369
- ewl_object_preferred_w_get, 370
- ewl_object_queued_add, 351
- ewl_object_queued_get, 351
- ewl_object_queued_has, 351
- ewl_object_queued_remove, 351
- ewl_object_recursive_get, 351
- ewl_object_recursive_set, 351
- ewl_object_size_request, 370
- ewl_object_state_add, 352
- ewl_object_state_get, 352
- ewl_object_state_has, 352
- ewl_object_state_remove, 352
- ewl_object_toplevel_get, 352
- ewl_object_toplevel_set, 353
- ewl_object_visible_add, 353
- ewl_object_visible_get, 353
- ewl_object_visible_has, 353
- ewl_object_visible_remove, 353
- ewl_object_w_request, 370
- ewl_object_x_request, 370
- ewl_object_y_request, 371
- flags, 710
- h, 710
- HIDDEN, 353
- INSET_BOTTOM, 354
- INSET_HORIZONTAL, 354
- INSET_LEFT, 354
- INSET_RIGHT, 354
- INSET_TOP, 354
- INSET_VERTICAL, 354
- insets, 710
- l, 710
- maximum, 710
- MAXIMUM_H, 354
- MAXIMUM_W, 354
- minimum, 710
- MINIMUM_H, 354
- MINIMUM_W, 354
- OBSCURED, 354
- pad, 711
- PADDING_BOTTOM, 355
- PADDING_HORIZONTAL, 355
- PADDING_LEFT, 355
- PADDING_RIGHT, 355
- PADDING_TOP, 355
- PADDING_VERTICAL, 355
- preferred, 711
- PREFERRED_H, 355
- PREFERRED_W, 355
- r, 711
- REALIZED, 355
- RECURSIVE, 355
- t, 711
- VISIBLE, 355
- w, 711
- x, 711
- y, 711
- ewl_object.c
 - ewl_object_maximum_inner_h_get, 1125
 - ewl_object_maximum_inner_w_get, 1126
 - ewl_object_minimum_inner_h_get, 1126
 - ewl_object_minimum_inner_w_get, 1126
- Ewl_Object: Base object. Provides functionality available to all widgets., 342
- ewl_object_alignment_get
 - Ewl_Object, 356
- ewl_object_alignment_set
 - Ewl_Object, 356
- ewl_object_current_geometry_get
 - Ewl_Object, 356
- ewl_object_current_h_get
 - Ewl_Object, 357
- ewl_object_current_size_get
 - Ewl_Object, 357
- ewl_object_current_w_get
 - Ewl_Object, 357
- ewl_object_current_x_get
 - Ewl_Object, 357
- ewl_object_current_y_get
 - Ewl_Object, 358
- ewl_object_custom_h_set
 - Ewl_Object, 348
- ewl_object_custom_size_set
 - Ewl_Object, 348
- ewl_object_custom_w_set
 - Ewl_Object, 349
- ewl_object_fill_policy_get
 - Ewl_Object, 358

- ewl_object_fill_policy_set
 - Ewl_Object, 358
- ewl_object_flags_add
 - Ewl_Object, 358
- ewl_object_flags_get
 - Ewl_Object, 349
- ewl_object_flags_has
 - Ewl_Object, 349
- ewl_object_flags_has_all
 - Ewl_Object, 350
- ewl_object_flags_remove
 - Ewl_Object, 359
- ewl_object_geometry_request
 - Ewl_Object, 359
- ewl_object_h_request
 - Ewl_Object, 359
- ewl_object_in_tab_list_get
 - Ewl_Object, 350
- ewl_object_in_tab_list_set
 - Ewl_Object, 350
- ewl_object_init
 - Ewl_Object, 360
- ewl_object_insets_bottom_get
 - Ewl_Object, 360
- ewl_object_insets_get
 - Ewl_Object, 360
- ewl_object_insets_left_get
 - Ewl_Object, 360
- ewl_object_insets_right_get
 - Ewl_Object, 361
- ewl_object_insets_set
 - Ewl_Object, 361
- ewl_object_insets_top_get
 - Ewl_Object, 361
- EWL_OBJECT_MAX_SIZE
 - Ewl_Object, 351
- ewl_object_maximum_h_get
 - Ewl_Object, 361
- ewl_object_maximum_h_set
 - Ewl_Object, 362
- ewl_object_maximum_inner_h_get
 - ewl_object.c, 1125
- ewl_object_maximum_inner_w_get
 - ewl_object.c, 1126
- ewl_object_maximum_size_get
 - Ewl_Object, 362
- ewl_object_maximum_size_set
 - Ewl_Object, 362
- ewl_object_maximum_w_get
 - Ewl_Object, 363
- ewl_object_maximum_w_set
 - Ewl_Object, 363
- EWL_OBJECT_MIN_SIZE
 - Ewl_Object, 351
- ewl_object_minimum_h_get
 - Ewl_Object, 363
- ewl_object_minimum_h_set
 - Ewl_Object, 363
- ewl_object_minimum_inner_h_get
 - ewl_object.c, 1126
- ewl_object_minimum_inner_w_get
 - ewl_object.c, 1126
- ewl_object_minimum_size_get
 - Ewl_Object, 364
- ewl_object_minimum_size_set
 - Ewl_Object, 364
- ewl_object_minimum_w_get
 - Ewl_Object, 364
- ewl_object_minimum_w_set
 - Ewl_Object, 365
- ewl_object_padding_bottom_get
 - Ewl_Object, 365
- ewl_object_padding_get
 - Ewl_Object, 365
- ewl_object_padding_left_get
 - Ewl_Object, 366
- ewl_object_padding_right_get
 - Ewl_Object, 366
- ewl_object_padding_set
 - Ewl_Object, 366
- ewl_object_padding_top_get
 - Ewl_Object, 366
- ewl_object_place
 - Ewl_Object, 367
- ewl_object_position_request
 - Ewl_Object, 367
- ewl_object_preferred_h_get
 - Ewl_Object, 367
- ewl_object_preferred_inner_h_get
 - Ewl_Object, 368
- ewl_object_preferred_inner_h_set
 - Ewl_Object, 368
- ewl_object_preferred_inner_size_get
 - Ewl_Object, 368
- ewl_object_preferred_inner_size_set
 - Ewl_Object, 368
- ewl_object_preferred_inner_w_get
 - Ewl_Object, 369
- ewl_object_preferred_inner_w_set
 - Ewl_Object, 369
- ewl_object_preferred_size_get
 - Ewl_Object, 369
- ewl_object_preferred_w_get
 - Ewl_Object, 370
- ewl_object_queued_add
 - Ewl_Object, 351
- ewl_object_queued_get
 - Ewl_Object, 351

- ewl_object_queued_has
 - Ewl_Object, 351
- ewl_object_queued_remove
 - Ewl_Object, 351
- ewl_object_recursive_get
 - Ewl_Object, 351
- ewl_object_recursive_set
 - Ewl_Object, 351
- ewl_object_size_request
 - Ewl_Object, 370
- ewl_object_state_add
 - Ewl_Object, 352
- ewl_object_state_get
 - Ewl_Object, 352
- ewl_object_state_has
 - Ewl_Object, 352
- ewl_object_state_remove
 - Ewl_Object, 352
- ewl_object_toplevel_get
 - Ewl_Object, 352
- ewl_object_toplevel_set
 - Ewl_Object, 353
- ewl_object_visible_add
 - Ewl_Object, 353
- ewl_object_visible_get
 - Ewl_Object, 353
- ewl_object_visible_has
 - Ewl_Object, 353
- ewl_object_visible_remove
 - Ewl_Object, 353
- ewl_object_w_request
 - Ewl_Object, 370
- ewl_object_x_request
 - Ewl_Object, 370
- ewl_object_y_request
 - Ewl_Object, 371
- Ewl_Orientation
 - Ewl_Enums, 152, 161
- EWL_ORIENTATION_HORIZONTAL
 - Ewl_Enums, 161
- EWL_ORIENTATION_VERTICAL
 - Ewl_Enums, 161
- EWL_OVERLAY
 - Ewl_Overlay, 372
- Ewl_Overlay, 712
 - container, 712
 - EWL_OVERLAY, 372
 - Ewl_Overlay, 373
 - ewl_overlay_cb_child_resize, 373
 - ewl_overlay_cb_child_show, 373
 - ewl_overlay_cb_configure, 373
 - ewl_overlay_init, 373
 - EWL_OVERLAY_IS, 372
 - ewl_overlay_new, 373
 - EWL_OVERLAY_TYPE, 373
- ewl_overlay.c
 - ewl_overlay_cb_configure, 1133
- Ewl_Overlay: A Container for Displaying on a canvas, 372
- ewl_overlay_cb_child_resize
 - Ewl_Overlay, 373
- ewl_overlay_cb_child_show
 - Ewl_Overlay, 373
- ewl_overlay_cb_configure
 - Ewl_Overlay, 373
 - ewl_overlay.c, 1133
- ewl_overlay_init
 - Ewl_Overlay, 373
- EWL_OVERLAY_IS
 - Ewl_Overlay, 372
- ewl_overlay_new
 - Ewl_Overlay, 373
- EWL_OVERLAY_TYPE
 - Ewl_Overlay, 373
- EWL_PAIR
 - Ewl_Widget, 545
- Ewl_Pair, 713
 - Ewl_Widget, 546
 - key, 713
 - value, 713
- Ewl_Pair_List, 714
 - direct, 714
 - Ewl_Widget, 546
 - len, 714
 - list, 714
- EWL_PANED
 - Ewl_Paned, 375
- Ewl_Paned, 715
 - container, 715
 - ewl_hpaned_new, 376
 - EWL_PANED, 375
 - Ewl_Paned, 376
 - ewl_paned_cb_child_add, 376
 - ewl_paned_cb_child_hide, 377
 - ewl_paned_cb_child_remove, 377
 - ewl_paned_cb_child_show, 377
 - ewl_paned_cb_configure, 377
 - ewl_paned_cb_destroy, 377
 - EWL_PANED_GRABBER, 375
 - Ewl_Paned_Grabber, 376
 - ewl_paned_grabber_init, 377
 - EWL_PANED_GRABBER_IS, 375
 - ewl_paned_grabber_new, 377
 - ewl_paned_grabber_orientation_get, 377
 - ewl_paned_grabber_paned_ - orientation_set, 377
 - ewl_paned_grabber_show_cursor_for, 377

- EWL_PANED_GRABBER_TYPE, 375
- ewl_paned_init, 377
- EWL_PANED_IS, 376
- ewl_paned_new, 377
- ewl_paned_orientation_get, 377
- ewl_paned_orientation_set, 378
- EWL_PANED_TYPE, 376
- ewl_vpaned_new, 378
- last_grab, 715
- last_pos, 715
- last_size, 715
- new_panes, 716
- orientation, 716
- updating_grabbers, 716
- Ewl_Paned: A paned widget, 374
- ewl_paned_cb_child_add
 - Ewl_Paned, 376
- ewl_paned_cb_child_hide
 - Ewl_Paned, 377
- ewl_paned_cb_child_remove
 - Ewl_Paned, 377
- ewl_paned_cb_child_show
 - Ewl_Paned, 377
- ewl_paned_cb_configure
 - Ewl_Paned, 377
 - lib/ewl_paned.c, 1136
- ewl_paned_cb_destroy
 - Ewl_Paned, 377
- EWL_PANED_GRABBER
 - Ewl_Paned, 375
- Ewl_Paned_Grabber, 717
 - Ewl_Paned, 376
 - placed, 717
 - separator, 717
- ewl_paned_grabber_init
 - Ewl_Paned, 377
- EWL_PANED_GRABBER_IS
 - Ewl_Paned, 375
- ewl_paned_grabber_new
 - Ewl_Paned, 377
- ewl_paned_grabber_orientation_get
 - Ewl_Paned, 377
- ewl_paned_grabber_paned_orientation_set
 - Ewl_Paned, 377
- ewl_paned_grabber_show_cursor_for
 - Ewl_Paned, 377
- EWL_PANED_GRABBER_TYPE
 - Ewl_Paned, 375
- ewl_paned_init
 - Ewl_Paned, 377
- EWL_PANED_IS
 - Ewl_Paned, 376
- Ewl_Paned_Layout, 718
 - current_position, 718
 - current_size, 718
 - minimum_size, 718
 - position_request, 718
 - preferred_size, 718
 - stable_position_request, 718
 - stable_request, 718
 - variable_request, 718
- ewl_paned_new
 - Ewl_Paned, 377
- ewl_paned_orientation_get
 - Ewl_Paned, 377
- ewl_paned_orientation_set
 - Ewl_Paned, 378
- EWL_PANED_TYPE
 - Ewl_Paned, 376
- EWL_PASSWORD
 - Ewl_Password, 380
- Ewl_Password, 719
 - entry, 719
 - EWL_PASSWORD, 380
 - Ewl_Password, 380
 - ewl_password_cb_destroy, 381
 - ewl_password_cb_key_down, 381
 - ewl_password_init, 381
 - EWL_PASSWORD_IS, 380
 - ewl_password_new, 381
 - ewl_password_obscure_get, 381
 - ewl_password_obscure_set, 381
 - ewl_password_text_get, 381
 - ewl_password_text_set, 382
 - EWL_PASSWORD_TYPE, 380
 - obscure, 719
 - real_text, 719
- Ewl_Password: A Single Line Text Password
 - Widget, 379
- ewl_password_cb_destroy
 - Ewl_Password, 381
 - lib/ewl_password.c, 1141
- ewl_password_cb_key_down
 - Ewl_Password, 381
 - lib/ewl_password.c, 1141
- ewl_password_init
 - Ewl_Password, 381
- EWL_PASSWORD_IS
 - Ewl_Password, 380
- ewl_password_new
 - Ewl_Password, 381
- ewl_password_obscure_get
 - Ewl_Password, 381
- ewl_password_obscure_set
 - Ewl_Password, 381
- ewl_password_text_get
 - Ewl_Password, 381
- ewl_password_text_set

- Ewl_Password, 382
- EWL_PASSWORD_TYPE
 - Ewl_Password, 380
- ewl_pointer.c
 - test_info, 813
- EWL_POPUP
 - Ewl_Popup, 384
- Ewl_Popup, 721
 - EWL_POPUP, 384
 - Ewl_Popup, 384
 - ewl_popup_cb_child_resize, 385
 - ewl_popup_cb_child_show, 385
 - ewl_popup_cb_destroy, 385
 - ewl_popup_cb_follow_configure, 385
 - ewl_popup_cb_follow_destroy, 385
 - ewl_popup_cb_show, 385
 - ewl_popup_fit_to_follow_get, 385
 - ewl_popup_fit_to_follow_set, 385
 - ewl_popup_follow_get, 385
 - ewl_popup_follow_set, 385
 - ewl_popup_init, 386
 - EWL_POPUP_IS, 384
 - ewl_popup_mouse_position_set, 386
 - ewl_popup_new, 386
 - ewl_popup_offset_set, 386
 - EWL_POPUP_TYPE, 384
 - ewl_popup_type_get, 387
 - ewl_popup_type_set, 387
 - fit_to_follow, 722
 - follow, 722
 - mouse, 722
 - offset, 722
 - type, 722
 - window, 722
 - x, 722
 - y, 722
- ewl_popup.c
 - ewl_popup_cb_destroy, 1146
 - ewl_popup_cb_follow_configure, 1146
 - ewl_popup_cb_follow_destroy, 1146
 - ewl_popup_cb_show, 1146
- Ewl_Popup: A override window for the convenience use, 383
- ewl_popup_cb_child_resize
 - Ewl_Popup, 385
- ewl_popup_cb_child_show
 - Ewl_Popup, 385
- ewl_popup_cb_destroy
 - Ewl_Popup, 385
 - ewl_popup.c, 1146
- ewl_popup_cb_follow_configure
 - Ewl_Popup, 385
 - ewl_popup.c, 1146
- ewl_popup_cb_follow_destroy
 - Ewl_Popup, 385
 - ewl_popup.c, 1146
- ewl_popup_fit_to_follow_get
 - Ewl_Popup, 385
- ewl_popup_fit_to_follow_set
 - Ewl_Popup, 385
- ewl_popup_follow_get
 - Ewl_Popup, 385
- ewl_popup_follow_set
 - Ewl_Popup, 385
- ewl_popup_init
 - Ewl_Popup, 386
- EWL_POPUP_IS
 - Ewl_Popup, 384
- ewl_popup_mouse_position_set
 - Ewl_Popup, 386
- ewl_popup_new
 - Ewl_Popup, 386
- ewl_popup_offset_set
 - Ewl_Popup, 386
- EWL_POPUP_TYPE
 - Ewl_Popup, 384
- Ewl_Popup_Type
 - Ewl_Enums, 152, 161
- ewl_popup_type_get
 - Ewl_Popup, 387
- EWL_POPUP_TYPE_MENU_HORIZONTAL
 - Ewl_Enums, 161
- EWL_POPUP_TYPE_MENU_VERTICAL
 - Ewl_Enums, 161
- EWL_POPUP_TYPE_MOUSE
 - Ewl_Enums, 161
- EWL_POPUP_TYPE_NONE
 - Ewl_Enums, 161
- ewl_popup_type_set
 - Ewl_Popup, 387
- Ewl_Position
 - Ewl_Enums, 152, 161
- EWL_POSITION_BOTTOM
 - Ewl_Enums, 161
- EWL_POSITION_LEFT
 - Ewl_Enums, 161
- EWL_POSITION_MASK
 - Ewl_Enums, 151
- EWL_POSITION_RIGHT
 - Ewl_Enums, 161
- EWL_POSITION_TOP
 - Ewl_Enums, 161
- ewl_print_help
 - Ewl_Misc, 314

- ewl_print_warning
 - ewl_debug.h, 904
 - ewl_misc.c, 1101
- ewl_private.h
 - __UNUSED__, 1153
 - ewl_config, 1153
 - EWL_CONFIG_CACHE_EVAS_-
FONT, 1153
 - EWL_CONFIG_CACHE_EVAS_-
IMAGE, 1153
 - EWL_CONFIG_DEBUG_-
BACKTRACE, 1153
 - EWL_CONFIG_DEBUG_ENABLE,
1153
 - EWL_CONFIG_DEBUG_EVAS_-
RENDER, 1153
 - EWL_CONFIG_DEBUG_GC_REAP,
1153
 - EWL_CONFIG_DEBUG_LEVEL, 1153
 - EWL_CONFIG_DEBUG_SEGV, 1153
 - EWL_CONFIG_ENGINE_NAME, 1153
 - EWL_CONFIG_THEME_COLOR_-
CLASSES_COUNT, 1153
 - EWL_CONFIG_THEME_COLOR_-
CLASSES_OVERRIDE, 1153
 - EWL_CONFIG_THEME_ICON_SIZE,
1153
 - EWL_CONFIG_THEME_ICON_-
THEME, 1153
 - EWL_CONFIG_THEME_NAME, 1153
 - EWL_CONFIG_THEME_PRINT_-
KEYS, 1153
 - EWL_CONFIG_THEME_PRINT_-
SIGNALS, 1153
 - PATH_MAX, 1153
- EWL_PROGRESSBAR
 - Ewl_Progressbar, 389
- Ewl_Progressbar, 723
 - auto_label, 723
 - bar, 723
 - EWL_PROGRESSBAR, 389
 - Ewl_Progressbar, 389
 - ewl_progressbar_cb_child_resize, 389
 - ewl_progressbar_cb_child_show, 389
 - ewl_progressbar_cb_configure, 389
 - ewl_progressbar_cb_value_changed, 389
 - ewl_progressbar_custom_label_set, 389
 - ewl_progressbar_init, 390
 - EWL_PROGRESSBAR_IS, 389
 - ewl_progressbar_label_hide, 390
 - ewl_progressbar_label_set, 390
 - ewl_progressbar_label_show, 390
 - ewl_progressbar_new, 390
 - EWL_PROGRESSBAR_TYPE, 389
- label, 723
 - range, 723
- Ewl_Progressbar: A progressbar from a value,
388
- ewl_progressbar_cb_child_resize
 - Ewl_Progressbar, 389
- ewl_progressbar_cb_child_show
 - Ewl_Progressbar, 389
- ewl_progressbar_cb_configure
 - Ewl_Progressbar, 389
 - lib/ewl_progressbar.c, 1155
- ewl_progressbar_cb_value_changed
 - Ewl_Progressbar, 389
 - lib/ewl_progressbar.c, 1155
- ewl_progressbar_custom_label_set
 - Ewl_Progressbar, 389
- ewl_progressbar_init
 - Ewl_Progressbar, 390
- EWL_PROGRESSBAR_IS
 - Ewl_Progressbar, 389
- ewl_progressbar_label_hide
 - Ewl_Progressbar, 390
- ewl_progressbar_label_set
 - Ewl_Progressbar, 390
- ewl_progressbar_label_show
 - Ewl_Progressbar, 390
- ewl_progressbar_new
 - Ewl_Progressbar, 390
- EWL_PROGRESSBAR_TYPE
 - Ewl_Progressbar, 389
- ewl_puzzle.c
 - test_info, 814
- EWL_RADIOBUTTON
 - Ewl_Radiobutton, 393
- Ewl_Radiobutton, 725
 - button, 725
 - chain, 725
 - EWL_RADIOBUTTON, 393
 - Ewl_Radiobutton, 393
 - ewl_radiobutton_cb_clicked, 393
 - ewl_radiobutton_cb_destroy, 393
 - ewl_radiobutton_chain_set, 393
 - ewl_radiobutton_checked_set, 393
 - ewl_radiobutton_init, 394
 - EWL_RADIOBUTTON_IS, 393
 - ewl_radiobutton_is_checked, 393
 - ewl_radiobutton_new, 394
 - EWL_RADIOBUTTON_TYPE, 393
- ewl_radiobutton.c
 - ewl_radiobutton_cb_clicked, 1159
 - ewl_radiobutton_cb_destroy, 1159
- Ewl_Radiobutton: A Radio Button Widget
and Grouping System, 392
- ewl_radiobutton_cb_clicked

- Ewl_Radiobutton, 393
- ewl_radiobutton.c, 1159
- ewl_radiobutton_cb_destroy
 - Ewl_Radiobutton, 393
 - ewl_radiobutton.c, 1159
- ewl_radiobutton_chain_set
 - Ewl_Radiobutton, 393
- ewl_radiobutton_checked_set
 - Ewl_Radiobutton, 393
- ewl_radiobutton_init
 - Ewl_Radiobutton, 394
- EWL_RADIOBUTTON_IS
 - Ewl_Radiobutton, 393
- ewl_radiobutton_is_checked
 - Ewl_Radiobutton, 393
- ewl_radiobutton_new
 - Ewl_Radiobutton, 394
- EWL_RADIOBUTTON_TYPE
 - Ewl_Radiobutton, 393
- EWL_RANGE
 - Ewl_Range, 396
- Ewl_Range, 727
 - container, 727
 - EWL_RANGE, 396
 - Ewl_Range, 396
 - ewl_range_decrease, 396
 - ewl_range_increase, 397
 - ewl_range_init, 397
 - ewl_range_invert_get, 397
 - ewl_range_invert_set, 397
 - EWL_RANGE_IS, 396
 - ewl_range_maximum_value_get, 398
 - ewl_range_maximum_value_set, 398
 - ewl_range_minimum_value_get, 398
 - ewl_range_minimum_value_set, 398
 - ewl_range_step_get, 399
 - ewl_range_step_set, 399
 - EWL_RANGE_TYPE, 396
 - ewl_range_value_get, 399
 - ewl_range_value_set, 399
 - invert, 727
 - max_val, 727
 - min_val, 727
 - step, 728
 - value, 728
- Ewl_Range: A Value Selector from a Range, 395
- ewl_range_decrease
 - Ewl_Range, 396
- ewl_range_increase
 - Ewl_Range, 397
- ewl_range_init
 - Ewl_Range, 397
- ewl_range_invert_get
 - Ewl_Range, 397
- ewl_range_invert_set
 - Ewl_Range, 397
- EWL_RANGE_IS
 - Ewl_Range, 396
- ewl_range_maximum_value_get
 - Ewl_Range, 398
- ewl_range_maximum_value_set
 - Ewl_Range, 398
- ewl_range_minimum_value_get
 - Ewl_Range, 398
- ewl_range_minimum_value_set
 - Ewl_Range, 398
- ewl_range_step_get
 - Ewl_Range, 399
- ewl_range_step_set
 - Ewl_Range, 399
- EWL_RANGE_TYPE
 - Ewl_Range, 396
- ewl_range_value_get
 - Ewl_Range, 399
- ewl_range_value_set
 - Ewl_Range, 399
- ewl_realize_cancel_request
 - Ewl_Misc, 314
- ewl_realize_request
 - Ewl_Misc, 314
- Ewl_Rotate
 - Ewl_Enums, 152, 161
- EWL_ROTATE_180
 - Ewl_Enums, 161
- EWL_ROTATE_CC_270
 - Ewl_Enums, 161
- EWL_ROTATE_CC_90
 - Ewl_Enums, 161
- EWL_ROTATE_CW_270
 - Ewl_Enums, 161
- EWL_ROTATE_CW_90
 - Ewl_Enums, 161
- EWL_ROW
 - Ewl_Row, 402
- Ewl_Row, 729
 - container, 729
 - EWL_ROW, 402
 - Ewl_Row, 402
 - ewl_row_cb_child_hide, 402
 - ewl_row_cb_child_resize, 402
 - ewl_row_cb_child_show, 402
 - ewl_row_cb_configure, 402
 - ewl_row_cb_header_configure, 402
 - ewl_row_cb_header_destroy, 402
 - ewl_row_column_get, 402
 - ewl_row_header_set, 402
 - ewl_row_init, 403

- EWL_ROW_IS, 402
- ewl_row_new, 403
- EWL_ROW_TYPE, 402
- header, 729
- ewl_row.c
 - ewl_row_cb_child_resize, 1167
 - ewl_row_cb_configure, 1167
 - ewl_row_cb_header_configure, 1167
 - ewl_row_cb_header_destroy, 1167
- Ewl_Row: The Row widget, 401
- ewl_row_cb_child_hide
 - Ewl_Row, 402
- ewl_row_cb_child_resize
 - Ewl_Row, 402
 - ewl_row.c, 1167
- ewl_row_cb_child_show
 - Ewl_Row, 402
- ewl_row_cb_configure
 - Ewl_Row, 402
 - ewl_row.c, 1167
- ewl_row_cb_header_configure
 - Ewl_Row, 402
 - ewl_row.c, 1167
- ewl_row_cb_header_destroy
 - Ewl_Row, 402
 - ewl_row.c, 1167
- ewl_row_column_get
 - Ewl_Row, 402
- ewl_row_header_set
 - Ewl_Row, 402
- ewl_row_init
 - Ewl_Row, 403
- EWL_ROW_IS
 - Ewl_Row, 402
- ewl_row_new
 - Ewl_Row, 403
- EWL_ROW_TYPE
 - Ewl_Row, 402
- EWL_SCROLLBAR
 - Ewl_Scrollbar, 405
- Ewl_Scrollbar, 730
 - box, 730
 - buttons_alignment, 730
 - decrement, 730
 - direction, 731
 - ewl_hscrollbar_new, 405
 - EWL_SCROLLBAR, 405
 - Ewl_Scrollbar, 405
 - ewl_scrollbar_cb_scroll_start, 406
 - ewl_scrollbar_cb_scroll_stop, 406
 - ewl_scrollbar_init, 406
 - ewl_scrollbar_inverse_scroll_get, 406
 - ewl_scrollbar_inverse_scroll_set, 406
 - EWL_SCROLLBAR_IS, 405
 - ewl_scrollbar_new, 406
 - ewl_scrollbar_orientation_get, 406
 - ewl_scrollbar_orientation_set, 407
 - ewl_scrollbar_step_get, 407
 - ewl_scrollbar_step_set, 407
 - EWL_SCROLLBAR_TYPE, 405
 - ewl_scrollbar_value_get, 407
 - ewl_scrollbar_value_set, 408
 - ewl_vscrollbar_new, 408
 - fill_percentage, 731
 - increment, 731
 - invert, 731
 - seeker, 731
 - start_time, 731
 - timer, 731
- Ewl_Scrollbar: A Simple Scrollbar Widget, 404
- ewl_scrollbar_cb_scroll_start
 - Ewl_Scrollbar, 406
 - lib/ewl_scrollbar.c, 1171
- ewl_scrollbar_cb_scroll_stop
 - Ewl_Scrollbar, 406
 - lib/ewl_scrollbar.c, 1171
- ewl_scrollbar_init
 - Ewl_Scrollbar, 406
- ewl_scrollbar_inverse_scroll_get
 - Ewl_Scrollbar, 406
- ewl_scrollbar_inverse_scroll_set
 - Ewl_Scrollbar, 406
- EWL_SCROLLBAR_IS
 - Ewl_Scrollbar, 405
- ewl_scrollbar_new
 - Ewl_Scrollbar, 406
- ewl_scrollbar_orientation_get
 - Ewl_Scrollbar, 406
- ewl_scrollbar_orientation_set
 - Ewl_Scrollbar, 407
- ewl_scrollbar_step_get
 - Ewl_Scrollbar, 407
- ewl_scrollbar_step_set
 - Ewl_Scrollbar, 407
- EWL_SCROLLBAR_TYPE
 - Ewl_Scrollbar, 405
- ewl_scrollbar_value_get
 - Ewl_Scrollbar, 407
- ewl_scrollbar_value_set
 - Ewl_Scrollbar, 408
- EWL_SCROLLPANE
 - Ewl_Scrollpane, 410
- Ewl_Scrollpane, 732
 - box, 732
 - container, 732
 - EWL_SCROLLPANE, 410
 - Ewl_Scrollpane, 411
 - ewl_scrollpane_cb_child_resize, 411

- ewl_scrollpane_cb_configure, 411
- ewl_scrollpane_cb_focus_jump, 411
- ewl_scrollpane_cb_hscroll, 411
- ewl_scrollpane_cb_vscroll, 411
- ewl_scrollpane_cb_wheel_scroll, 411
- ewl_scrollpane_hscrollbar_flag_get, 411
- ewl_scrollpane_hscrollbar_flag_set, 411
- ewl_scrollpane_hscrollbar_step_get, 411
- ewl_scrollpane_hscrollbar_value_get, 412
- ewl_scrollpane_hscrollbar_value_set, 412
- ewl_scrollpane_init, 412
- EWL_SCROLLPANE_IS, 410
- ewl_scrollpane_new, 412
- EWL_SCROLLPANE_TYPE, 410
- ewl_scrollpane_vscrollbar_flag_get, 413
- ewl_scrollpane_vscrollbar_flag_set, 413
- ewl_scrollpane_vscrollbar_step_get, 413
- ewl_scrollpane_vscrollbar_value_get, 413
- ewl_scrollpane_vscrollbar_value_set, 414
- hflag, 732
- hscrollbar, 732
- overlay, 733
- vflag, 733
- vscrollbar, 733
- Ewl_Scrollpane: A Scrollable Viewing Area, 409
- ewl_scrollpane_cb_child_resize
 - Ewl_Scrollpane, 411
- ewl_scrollpane_cb_configure
 - Ewl_Scrollpane, 411
 - lib/ewl_scrollpane.c, 1176
- ewl_scrollpane_cb_focus_jump
 - Ewl_Scrollpane, 411
 - lib/ewl_scrollpane.c, 1176
- ewl_scrollpane_cb_hscroll
 - Ewl_Scrollpane, 411
 - lib/ewl_scrollpane.c, 1176
- ewl_scrollpane_cb_vscroll
 - Ewl_Scrollpane, 411
 - lib/ewl_scrollpane.c, 1176
- ewl_scrollpane_cb_wheel_scroll
 - Ewl_Scrollpane, 411
 - lib/ewl_scrollpane.c, 1176
- EWL_SCROLLPANE_FLAG_ALWAYS_HIDDEN
 - Ewl_Enums, 162
- EWL_SCROLLPANE_FLAG_AUTO_VISIBLE
 - Ewl_Enums, 162
- EWL_SCROLLPANE_FLAG_NONE
 - Ewl_Enums, 162
- Ewl_Scrollpane_Flags
 - Ewl_Enums, 152, 161
- ewl_scrollpane_hscrollbar_flag_get
 - Ewl_Scrollpane, 411
- ewl_scrollpane_hscrollbar_flag_set
 - Ewl_Scrollpane, 411
- ewl_scrollpane_hscrollbar_step_get
 - Ewl_Scrollpane, 411
- ewl_scrollpane_hscrollbar_value_get
 - Ewl_Scrollpane, 412
- ewl_scrollpane_hscrollbar_value_set
 - Ewl_Scrollpane, 412
- ewl_scrollpane_init
 - Ewl_Scrollpane, 412
- EWL_SCROLLPANE_IS
 - Ewl_Scrollpane, 410
- ewl_scrollpane_new
 - Ewl_Scrollpane, 412
- EWL_SCROLLPANE_TYPE
 - Ewl_Scrollpane, 410
- ewl_scrollpane_vscrollbar_flag_get
 - Ewl_Scrollpane, 413
- ewl_scrollpane_vscrollbar_flag_set
 - Ewl_Scrollpane, 413
- ewl_scrollpane_vscrollbar_step_get
 - Ewl_Scrollpane, 413
- ewl_scrollpane_vscrollbar_value_get
 - Ewl_Scrollpane, 413
- ewl_scrollpane_vscrollbar_value_set
 - Ewl_Scrollpane, 414
- EWL_SEEKER
 - Ewl_Seeker, 416
- Ewl_Seeker, 734
 - autohide, 734
 - button, 734
 - dragstart, 734
 - ewl_hseeker_new, 416
 - EWL_SEEKER, 416
 - Ewl_Seeker, 416
 - ewl_seeker_autohide_get, 416
 - ewl_seeker_autohide_set, 417
 - ewl_seeker_cb_button_mouse_down, 417
 - ewl_seeker_cb_button_mouse_up, 417
 - ewl_seeker_cb_child_show, 417
 - ewl_seeker_cb_configure, 417
 - ewl_seeker_cb_key_down, 417
 - ewl_seeker_cb_mouse_down, 417
 - ewl_seeker_cb_mouse_move, 417
 - ewl_seeker_cb_mouse_up, 417
 - ewl_seeker_init, 417
 - EWL_SEEKER_IS, 416
 - ewl_seeker_new, 418
 - ewl_seeker_orientation_get, 418
 - ewl_seeker_orientation_set, 418

- EWL_SEEKER_TYPE, 416
- ewl_vseeker_new, 418
- orientation, 735
- range, 735
- start_time, 735
- timer, 735
- Ewl_Seeker: A Value Selector from a Range, 415
- ewl_seeker_autohide_get
 - Ewl_Seeker, 416
- ewl_seeker_autohide_set
 - Ewl_Seeker, 417
- ewl_seeker_cb_button_mouse_down
 - Ewl_Seeker, 417
 - lib/ewl_seeker.c, 1181
- ewl_seeker_cb_button_mouse_up
 - Ewl_Seeker, 417
 - lib/ewl_seeker.c, 1181
- ewl_seeker_cb_child_show
 - Ewl_Seeker, 417
- ewl_seeker_cb_configure
 - Ewl_Seeker, 417
 - lib/ewl_seeker.c, 1181
- ewl_seeker_cb_key_down
 - Ewl_Seeker, 417
 - lib/ewl_seeker.c, 1181
- ewl_seeker_cb_mouse_down
 - Ewl_Seeker, 417
 - lib/ewl_seeker.c, 1181
- ewl_seeker_cb_mouse_move
 - Ewl_Seeker, 417
 - lib/ewl_seeker.c, 1181
- ewl_seeker_cb_mouse_up
 - Ewl_Seeker, 417
 - lib/ewl_seeker.c, 1181
- ewl_seeker_init
 - Ewl_Seeker, 417
- EWL_SEEKER_IS
 - Ewl_Seeker, 416
- ewl_seeker_new
 - Ewl_Seeker, 418
- ewl_seeker_orientation_get
 - Ewl_Seeker, 418
- ewl_seeker_orientation_set
 - Ewl_Seeker, 418
- EWL_SEEKER_TYPE
 - Ewl_Seeker, 416
- ewl_segv
 - ewl_debug.h, 904
 - ewl_misc.c, 1101
- EWL_SELECTION
 - Ewl_MVC, 328
- Ewl_Selection, 736
 - data, 736
 - Ewl_MVC, 329
 - highlight, 736
 - model, 736
 - type, 736
- EWL_SELECTION_IDX
 - Ewl_MVC, 328
- Ewl_Selection_Idx, 737
 - column, 737
 - Ewl_MVC, 329
 - row, 737
 - sel, 737
- Ewl_Selection_Mode
 - Ewl_Enums, 153, 162
- EWL_SELECTION_MODE_MULTI
 - Ewl_Enums, 162
- EWL_SELECTION_MODE_NONE
 - Ewl_Enums, 162
- EWL_SELECTION_MODE_SINGLE
 - Ewl_Enums, 162
- EWL_SELECTION_RANGE
 - Ewl_MVC, 329
- Ewl_Selection_Range, 738
 - column, 738
 - end, 738
 - Ewl_MVC, 329
 - row, 738
 - sel, 738
 - start, 739
- Ewl_Selection_Type
 - Ewl_Enums, 153, 162
- EWL_SELECTION_TYPE_INDEX
 - Ewl_Enums, 162
- EWL_SELECTION_TYPE_RANGE
 - Ewl_Enums, 162
- EWL_SEPARATOR
 - Ewl_Separator, 420
- Ewl_Separator, 740
 - ewl_hseparator_new, 420
 - EWL_SEPARATOR, 420
 - Ewl_Separator, 420
 - ewl_separator_init, 420
 - EWL_SEPARATOR_IS, 420
 - ewl_separator_new, 420
 - ewl_separator_orientation_get, 420
 - ewl_separator_orientation_set, 421
 - EWL_SEPARATOR_TYPE, 420
 - ewl_vseparator_new, 421
 - orientation, 740
 - widget, 740
- Ewl_Separator: A Visual Separator Between Widgets, 419
- ewl_separator_init
 - Ewl_Separator, 420
- EWL_SEPARATOR_IS

- Ewl_Separator, 420
- ewl_separator_new
 - Ewl_Separator, 420
- ewl_separator_orientation_get
 - Ewl_Separator, 420
- ewl_separator_orientation_set
 - Ewl_Separator, 421
- EWL_SEPARATOR_TYPE
 - Ewl_Separator, 420
- EWL_SHADOW
 - Ewl_Shadow, 422
- Ewl_Shadow, 741
 - box, 741
 - EWL_SHADOW, 422
 - Ewl_Shadow, 423
 - ewl_shadow_init, 423
 - EWL_SHADOW_IS, 422
 - ewl_shadow_new, 423
 - EWL_SHADOW_TYPE, 422
- Ewl_Shadow: A container with a shadow/border, 422
- ewl_shadow_init
 - Ewl_Shadow, 423
- EWL_SHADOW_IS
 - Ewl_Shadow, 422
- ewl_shadow_new
 - Ewl_Shadow, 423
- EWL_SHADOW_TYPE
 - Ewl_Shadow, 422
- ewl_shutdown
 - Ewl_Misc, 314
- ewl_shutdown_add
 - Ewl_Misc, 314
- Ewl_Shutdown_Hook
 - Ewl_Misc, 313
- ewl_simple_test.c
 - __UNUSED__, 797
 - main, 797
- Ewl_Sort_Direction
 - Ewl_Enums, 153, 162
- EWL_SORT_DIRECTION_ASCENDING
 - Ewl_Enums, 162
- EWL_SORT_DIRECTION_DESCENDING
 - Ewl_Enums, 162
- EWL_SORT_DIRECTION_MAX
 - Ewl_Enums, 162
- EWL_SORT_DIRECTION_NONE
 - Ewl_Enums, 162
- EWL_SPACER
 - Ewl_Spacer, 424
- Ewl_Spacer, 742
 - EWL_SPACER, 424
 - Ewl_Spacer, 425
 - ewl_spacer_init, 425
- EWL_SPACER_IS, 424
- ewl_spacer_new, 425
- EWL_SPACER_TYPE, 424
- widget, 742
- Ewl_Spacer: A Widget to Add Space Between Other Widgets, 424
- ewl_spacer_init
 - Ewl_Spacer, 425
- EWL_SPACER_IS
 - Ewl_Spacer, 424
- ewl_spacer_new
 - Ewl_Spacer, 425
- EWL_SPACER_TYPE
 - Ewl_Spacer, 424
- EWL_SPECTRUM
 - Ewl_Spectrum, 427
- Ewl_Spectrum, 743
 - canvas, 744
 - cross_hairs, 744
 - dirty, 744
 - EWL_SPECTRUM, 427
 - Ewl_Spectrum, 427
 - ewl_spectrum_canvas_cb_reveal, 428
 - ewl_spectrum_cb_configure, 428
 - ewl_spectrum_cb_mouse_down, 428
 - ewl_spectrum_cb_mouse_move, 428
 - ewl_spectrum_cb_mouse_up, 428
 - ewl_spectrum_hsv_get, 428
 - ewl_spectrum_hsv_set, 428
 - ewl_spectrum_init, 428
 - EWL_SPECTRUM_IS, 427
 - ewl_spectrum_mode_get, 429
 - ewl_spectrum_mode_set, 429
 - ewl_spectrum_new, 429
 - ewl_spectrum_rgb_get, 429
 - ewl_spectrum_rgb_set, 430
 - EWL_SPECTRUM_TYPE, 427
 - ewl_spectrum_type_get, 430
 - ewl_spectrum_type_set, 430
 - h, 744
 - horizontal, 744
 - hsv, 744
 - mode, 744
 - overlay, 744
 - rgb, 744
 - s, 744
 - type, 744
 - v, 744
 - vertical, 745
- ewl_spectrum.c
 - ewl_spectrum_canvas_cb_reveal, 1194
 - ewl_spectrum_cb_configure, 1194
 - ewl_spectrum_cb_mouse_down, 1194
 - ewl_spectrum_cb_mouse_move, 1194

- ewl_spectrum_cb_mouse_up, 1194
- Ewl_Spectrum: The colour spectrum widget, 426
 - ewl_spectrum_canvas_cb_reveal
 - Ewl_Spectrum, 428
 - ewl_spectrum.c, 1194
 - ewl_spectrum_cb_configure
 - Ewl_Spectrum, 428
 - ewl_spectrum.c, 1194
 - ewl_spectrum_cb_mouse_down
 - Ewl_Spectrum, 428
 - ewl_spectrum.c, 1194
 - ewl_spectrum_cb_mouse_move
 - Ewl_Spectrum, 428
 - ewl_spectrum.c, 1194
 - ewl_spectrum_cb_mouse_up
 - Ewl_Spectrum, 428
 - ewl_spectrum.c, 1194
 - ewl_spectrum_hsv_get
 - Ewl_Spectrum, 428
 - ewl_spectrum_hsv_set
 - Ewl_Spectrum, 428
 - ewl_spectrum_init
 - Ewl_Spectrum, 428
- EWL_SPECTRUM_IS
 - Ewl_Spectrum, 427
- ewl_spectrum_mode_get
 - Ewl_Spectrum, 429
- ewl_spectrum_mode_set
 - Ewl_Spectrum, 429
- ewl_spectrum_new
 - Ewl_Spectrum, 429
- ewl_spectrum_rgb_get
 - Ewl_Spectrum, 429
- ewl_spectrum_rgb_set
 - Ewl_Spectrum, 430
- EWL_SPECTRUM_TYPE
 - Ewl_Spectrum, 427
- Ewl_Spectrum_Type
 - Ewl_Enums, 153, 162
- ewl_spectrum_type_get
 - Ewl_Spectrum, 430
- ewl_spectrum_type_set
 - Ewl_Spectrum, 430
- EWL_SPECTRUM_TYPE_SQUARE
 - Ewl_Enums, 162
- EWL_SPECTRUM_TYPE_VERTICAL
 - Ewl_Enums, 162
- EWL_SPINNER
 - Ewl_Spinner, 433
- Ewl_Spinner, 746
 - decrement, 746
 - digits, 746
 - direction, 747
 - entry, 747
- EWL_SPINNER, 433
 - Ewl_Spinner, 433
 - ewl_spinner_cb_child_resize, 434
 - ewl_spinner_cb_child_show, 434
 - ewl_spinner_cb_configure, 434
 - ewl_spinner_cb_decrease_value, 434
 - ewl_spinner_cb_destroy, 434
 - ewl_spinner_cb_focus_out, 434
 - ewl_spinner_cb_increase_value, 434
 - ewl_spinner_cb_key_down, 434
 - ewl_spinner_cb_realize, 434
 - ewl_spinner_cb_value_changed, 434
 - ewl_spinner_cb_value_stop, 434
 - ewl_spinner_cb_wheel, 434
 - ewl_spinner_digits_get, 434
 - ewl_spinner_digits_set, 434
 - ewl_spinner_init, 435
- EWL_SPINNER_IS, 433
 - ewl_spinner_new, 435
- EWL_SPINNER_TYPE, 433
 - increment, 747
 - last_value, 747
 - range, 747
 - start_time, 747
 - timer, 747
 - vbox, 747
- Ewl_Spinner: A Numerical Value Entry, 432
 - ewl_spinner_cb_child_resize
 - Ewl_Spinner, 434
 - ewl_spinner_cb_child_show
 - Ewl_Spinner, 434
 - ewl_spinner_cb_configure
 - Ewl_Spinner, 434
 - lib/ewl_spinner.c, 1198
 - ewl_spinner_cb_decrease_value
 - Ewl_Spinner, 434
 - ewl_spinner_cb_destroy
 - Ewl_Spinner, 434
 - lib/ewl_spinner.c, 1198
 - ewl_spinner_cb_focus_out
 - Ewl_Spinner, 434
 - lib/ewl_spinner.c, 1198
 - ewl_spinner_cb_increase_value
 - Ewl_Spinner, 434
 - ewl_spinner_cb_key_down
 - Ewl_Spinner, 434
 - lib/ewl_spinner.c, 1198
 - ewl_spinner_cb_realize
 - Ewl_Spinner, 434
 - lib/ewl_spinner.c, 1198
 - ewl_spinner_cb_value_changed
 - Ewl_Spinner, 434
 - lib/ewl_spinner.c, 1198

- ewl_spinner_cb_value_stop
 - Ewl_Spinner, 434
 - lib/ewl_spinner.c, 1198
- ewl_spinner_cb_wheel
 - Ewl_Spinner, 434
 - lib/ewl_spinner.c, 1198
- ewl_spinner_digits_get
 - Ewl_Spinner, 434
- ewl_spinner_digits_set
 - Ewl_Spinner, 434
- ewl_spinner_init
 - Ewl_Spinner, 435
- EWL_SPINNER_IS
 - Ewl_Spinner, 433
- ewl_spinner_new
 - Ewl_Spinner, 435
- EWL_SPINNER_TYPE
 - Ewl_Spinner, 433
- EWL_STATE_PERSISTENT
 - Ewl_Enums, 163
- EWL_STATE_TRANSIENT
 - Ewl_Enums, 163
- Ewl_State_Type
 - Ewl_Enums, 153, 162
- EWL_STATUSBAR
 - Ewl_Statusbar, 437
- Ewl_Statusbar, 748
 - current, 748
 - EWL_STATUSBAR, 437
 - Ewl_Statusbar, 437
 - ewl_statusbar_active_set, 437
 - ewl_statusbar_cb_destroy, 438
 - ewl_statusbar_init, 438
 - EWL_STATUSBAR_IS, 437
 - ewl_statusbar_left_append, 438
 - ewl_statusbar_left_hide, 438
 - ewl_statusbar_left_prepend, 438
 - ewl_statusbar_left_show, 439
 - ewl_statusbar_new, 439
 - ewl_statusbar_pop, 439
 - ewl_statusbar_push, 439
 - ewl_statusbar_right_append, 440
 - ewl_statusbar_right_hide, 440
 - ewl_statusbar_right_prepend, 440
 - ewl_statusbar_right_show, 440
 - EWL_STATUSBAR_TYPE, 437
 - left, 748
 - outer_box, 748
 - right, 749
 - stack, 749
 - status, 749
- Ewl_Statusbar: A status bar widget, 436
- ewl_statusbar_active_set
 - Ewl_Statusbar, 437
- ewl_statusbar_cb_destroy
 - Ewl_Statusbar, 438
 - lib/ewl_statusbar.c, 1203
- ewl_statusbar_init
 - Ewl_Statusbar, 438
- EWL_STATUSBAR_IS
 - Ewl_Statusbar, 437
- ewl_statusbar_left_append
 - Ewl_Statusbar, 438
- ewl_statusbar_left_hide
 - Ewl_Statusbar, 438
- ewl_statusbar_left_prepend
 - Ewl_Statusbar, 438
- ewl_statusbar_left_show
 - Ewl_Statusbar, 439
- ewl_statusbar_new
 - Ewl_Statusbar, 439
- ewl_statusbar_pop
 - Ewl_Statusbar, 439
- ewl_statusbar_push
 - Ewl_Statusbar, 439
- ewl_statusbar_right_append
 - Ewl_Statusbar, 440
- ewl_statusbar_right_hide
 - Ewl_Statusbar, 440
- ewl_statusbar_right_prepend
 - Ewl_Statusbar, 440
- ewl_statusbar_right_show
 - Ewl_Statusbar, 440
- EWL_STATUSBAR_TYPE
 - Ewl_Statusbar, 437
- EWL_STOCK
 - Ewl_Stock, 443
- Ewl_Stock, 750
 - box, 750
 - EWL_STOCK, 443
 - Ewl_Stock, 443
 - Ewl_Stock_Funcs, 443
 - ewl_stock_functions_set, 444
 - EWL_STOCK_IMAGE_SET, 443
 - Ewl_Stock_Image_Set, 443
 - ewl_stock_init, 444
 - EWL_STOCK_IS, 443
 - EWL_STOCK_LABEL_SET, 443
 - Ewl_Stock_Label_Set, 443
 - EWL_STOCK_TOOLTIP_SET, 443
 - Ewl_Stock_Tooltip_Set, 443
 - EWL_STOCK_TYPE, 443
 - ewl_stock_type_get, 444
 - ewl_stock_type_set, 444
 - stock_funcs, 750
 - stock_type, 750
- ewl_stock.c
 - ewl_stock_items, 1208

- image_key, 1208
- label, 1208
- tooltip, 1208
- Ewl_Stock: The Stock, 442
- EWL_STOCK_APPLY
 - Ewl_Enums, 163
- EWL_STOCK_ARROW_DOWN
 - Ewl_Enums, 163
- EWL_STOCK_ARROW_LEFT
 - Ewl_Enums, 163
- EWL_STOCK_ARROW_RIGHT
 - Ewl_Enums, 163
- EWL_STOCK_ARROW_UP
 - Ewl_Enums, 163
- EWL_STOCK_CANCEL
 - Ewl_Enums, 163
- EWL_STOCK_FASTFORWARD
 - Ewl_Enums, 163
- Ewl_Stock_Funcs, 752
 - Ewl_Stock, 443
 - image_set, 752
 - label_set, 752
 - tooltip_set, 752
- ewl_stock_functions_set
 - Ewl_Stock, 444
- EWL_STOCK_HOME
 - Ewl_Enums, 163
- EWL_STOCK_IMAGE_SET
 - Ewl_Stock, 443
- Ewl_Stock_Image_Set
 - Ewl_Stock, 443
- ewl_stock_init
 - Ewl_Stock, 444
- EWL_STOCK_IS
 - Ewl_Stock, 443
- ewl_stock_items
 - ewl_stock.c, 1208
- EWL_STOCK_LABEL_SET
 - Ewl_Stock, 443
- Ewl_Stock_Label_Set
 - Ewl_Stock, 443
- EWL_STOCK_NONE
 - Ewl_Enums, 163
- EWL_STOCK_OK
 - Ewl_Enums, 163
- EWL_STOCK_OPEN
 - Ewl_Enums, 163
- EWL_STOCK_PAUSE
 - Ewl_Enums, 163
- EWL_STOCK_PLAY
 - Ewl_Enums, 163
- EWL_STOCK_QUIT
 - Ewl_Enums, 163
- EWL_STOCK_REWIND
 - Ewl_Enums, 163
- EWL_STOCK_SAVE
 - Ewl_Enums, 163
- EWL_STOCK_STOP
 - Ewl_Enums, 163
- EWL_STOCK_TOOLTIP_SET
 - Ewl_Stock, 443
- Ewl_Stock_Tooltip_Set
 - Ewl_Stock, 443
- EWL_STOCK_TYPE
 - Ewl_Stock, 443
- Ewl_Stock_Type
 - Ewl_Enums, 153, 163
- ewl_stock_type_get
 - Ewl_Stock, 444
- ewl_stock_type_set
 - Ewl_Stock, 444
- EWL_TABLE
 - Ewl_Table, 446
- Ewl_Table, 753
 - col_headers, 753
 - container, 753
 - end_c, 754
 - end_r, 754
 - EWL_TABLE, 446
 - Ewl_Table, 447
 - ewl_table_add, 447
 - ewl_table_cb_child_select, 447
 - ewl_table_cb_child_show, 447
 - ewl_table_cb_configure, 447
 - ewl_table_col_row_get, 447
 - ewl_table_column_w_get, 447
 - ewl_table_column_w_set, 448
 - ewl_table_find, 448
 - ewl_table_hhomogeneous_get, 448
 - ewl_table_hhomogeneous_set, 449
 - ewl_table_homogeneous_set, 449
 - ewl_table_init, 449
 - EWL_TABLE_IS, 446
 - ewl_table_new, 450
 - ewl_table_reset, 450
 - ewl_table_row_h_get, 450
 - ewl_table_row_h_set, 450
 - ewl_table_selected_get, 451
 - EWL_TABLE_TYPE, 446
 - ewl_table_vhomogeneous_get, 451
 - ewl_table_vhomogeneous_set, 451
 - grid, 754
 - homogeneous_h, 754
 - homogeneous_v, 754
 - row_select, 754
 - selected, 754
 - start_c, 754
 - start_r, 754

- Ewl_Table: The Table Layout Container., 445
- ewl_table_add
 - Ewl_Table, 447
- ewl_table_cb_child_select
 - Ewl_Table, 447
 - lib/ewl_table.c, 1212
- ewl_table_cb_child_show
 - Ewl_Table, 447
 - lib/ewl_table.c, 1212
- ewl_table_cb_configure
 - Ewl_Table, 447
 - lib/ewl_table.c, 1212
- ewl_table_col_row_get
 - Ewl_Table, 447
- ewl_table_column_w_get
 - Ewl_Table, 447
- ewl_table_column_w_set
 - Ewl_Table, 448
- ewl_table_find
 - Ewl_Table, 448
- ewl_table_hhomogeneous_get
 - Ewl_Table, 448
- ewl_table_hhomogeneous_set
 - Ewl_Table, 449
- ewl_table_homogeneous_set
 - Ewl_Table, 449
- ewl_table_init
 - Ewl_Table, 449
- EWL_TABLE_IS
 - Ewl_Table, 446
- ewl_table_new
 - Ewl_Table, 450
- ewl_table_reset
 - Ewl_Table, 450
- ewl_table_row_h_get
 - Ewl_Table, 450
- ewl_table_row_h_set
 - Ewl_Table, 450
- ewl_table_selected_get
 - Ewl_Table, 451
- EWL_TABLE_TYPE
 - Ewl_Table, 446
- ewl_table_vhomogeneous_get
 - Ewl_Table, 451
- ewl_table_vhomogeneous_set
 - Ewl_Table, 451
- Ewl_Test, 755
 - Ewl_Test.h, 800
 - filename, 755
 - func, 755
 - handle, 755
 - name, 755
 - tip, 755
 - type, 755
 - unit_tests, 756
- Ewl_Test.h
 - Ewl_Test, 800
 - Ewl_Test_Type, 800
 - EWL_TEST_TYPE_ADVANCED, 800
 - EWL_TEST_TYPE_CONTAINER, 800
 - EWL_TEST_TYPE_MISC, 800
 - EWL_TEST_TYPE_SIMPLE, 800
 - EWL_TEST_TYPE_UNIT, 800
 - Ewl_Unit_Test, 800
- ewl_test_private.h
 - __UNUSED__, 803
- Ewl_Test_Type
 - Ewl_Test.h, 800
- EWL_TEST_TYPE_ADVANCED
 - Ewl_Test.h, 800
- EWL_TEST_TYPE_CONTAINER
 - Ewl_Test.h, 800
- EWL_TEST_TYPE_MISC
 - Ewl_Test.h, 800
- EWL_TEST_TYPE_SIMPLE
 - Ewl_Test.h, 800
- EWL_TEST_TYPE_UNIT
 - Ewl_Test.h, 800
- EWL_TEXT
 - Ewl_Text, 460
- Ewl_Text, 757
 - byte_idx, 758
 - bytes, 758
 - char_idx, 758
 - chars, 758
 - container, 758
 - current, 758
 - cursor_position, 758
 - delete_count, 758
 - dirty, 758
 - EWL_TEXT, 460
 - Ewl_Text, 461
 - ewl_text_align_apply, 461
 - ewl_text_align_get, 461
 - ewl_text_align_set, 461
 - ewl_text_bg_color_apply, 462
 - ewl_text_bg_color_get, 462
 - ewl_text_bg_color_set, 462
 - ewl_text_cb_child_add, 463
 - ewl_text_cb_child_del, 463
 - ewl_text_cb_configure, 463
 - ewl_text_cb_destroy, 463
 - ewl_text_cb_hide, 463
 - ewl_text_cb_mouse_down, 463
 - ewl_text_cb_mouse_move, 463
 - ewl_text_cb_mouse_up, 463
 - ewl_text_cb_obscure, 463
 - ewl_text_cb_reveal, 463

ewl_text_cb_show, 463
ewl_text_clear, 463
ewl_text_color_apply, 463
ewl_text_color_get, 464
ewl_text_color_set, 464
Ewl_Text_Context, 461
ewl_text_context_acquire, 464
ewl_text_context_compare, 465
ewl_text_context_default_create, 465
ewl_text_context_dup, 465
ewl_text_context_init, 465
ewl_text_context_new, 465
ewl_text_context_release, 465
ewl_text_context_shutdown, 465
ewl_text_coord_index_map, 465
ewl_text_cursor_position_get, 465
ewl_text_cursor_position_line_down_-
get, 465
ewl_text_cursor_position_line_up_get,
466
ewl_text_cursor_position_set, 466
ewl_text_double_underline_color_-
apply, 466
ewl_text_double_underline_color_get,
467
ewl_text_double_underline_color_set,
467
ewl_text_fmt_dump, 467
ewl_text_font_apply, 467
ewl_text_font_get, 468
ewl_text_font_set, 468
ewl_text_font_size_apply, 468
ewl_text_font_size_get, 468
ewl_text_font_size_set, 469
ewl_text_font_source_apply, 469
ewl_text_font_source_get, 469
ewl_text_font_source_set, 469
ewl_text_glow_color_apply, 470
ewl_text_glow_color_get, 470
ewl_text_glow_color_set, 470
ewl_text_has_selection, 471
ewl_text_index_geometry_map, 471
ewl_text_init, 471
EWL_TEXT_IS, 460
ewl_text_length_get, 471
ewl_text_new, 472
ewl_text_offsets_get, 472
ewl_text_offsets_set, 472
ewl_text_outline_color_apply, 472
ewl_text_outline_color_get, 473
ewl_text_outline_color_set, 473
ewl_text_selectable_get, 473
ewl_text_selectable_set, 474
ewl_text_selection_get, 474
EWL_TEXT_SELECTION_IS, 460
ewl_text_selection_text_get, 474
EWL_TEXT_SELECTION_TYPE, 460
ewl_text_shadow_color_apply, 474
ewl_text_shadow_color_get, 475
ewl_text_shadow_color_set, 475
ewl_text_strikethrough_color_apply, 475
ewl_text_strikethrough_color_get, 476
ewl_text_strikethrough_color_set, 476
ewl_text_style_add, 477
ewl_text_style_del, 477
ewl_text_style_has, 477
ewl_text_style_invert, 477
ewl_text_styles_apply, 478
ewl_text_styles_get, 478
ewl_text_styles_set, 478
ewl_text_text_append, 478
ewl_text_text_delete, 479
ewl_text_text_get, 479
ewl_text_text_insert, 479
ewl_text_text_prepend, 479
ewl_text_text_set, 480
EWL_TEXT_TRIGGER, 460
Ewl_Text_Trigger, 461
EWL_TEXT_TRIGGER_AREA, 460
Ewl_Text_Trigger_Area, 461
ewl_text_trigger_area_init, 480
ewl_text_trigger_area_new, 480
ewl_text_trigger_base_get, 480
ewl_text_trigger_base_set, 480
ewl_text_trigger_cb_destroy, 481
ewl_text_trigger_cb_mouse_down, 481
ewl_text_trigger_cb_mouse_in, 481
ewl_text_trigger_cb_mouse_out, 481
ewl_text_trigger_cb_mouse_up, 481
ewl_text_trigger_init, 481
EWL_TEXT_TRIGGER_IS, 460
ewl_text_trigger_length_get, 481
ewl_text_trigger_length_set, 481
ewl_text_trigger_new, 482
ewl_text_trigger_start_pos_get, 482
ewl_text_trigger_start_pos_set, 482
EWL_TEXT_TRIGGER_TYPE, 460
ewl_text_trigger_type_get, 482
ewl_text_triggers_configure, 483
ewl_text_triggers_hide, 483
ewl_text_triggers_realize, 483
ewl_text_triggers_show, 483
ewl_text_triggers_unrealize, 483
EWL_TEXT_TYPE, 460
ewl_text_underline_color_apply, 483
ewl_text_underline_color_get, 483
ewl_text_underline_color_set, 484
ewl_text_wrap_apply, 484

- ewl_text_wrap_get, 484
- ewl_text_wrap_set, 484
- formatting, 759
- in_select, 759
- length, 759
- nodes, 759
- offset, 759
- selectable, 759
- selection, 759
- text, 759
- textblock, 759
- total_size, 759
- triggers, 759
- tx, 759
- x, 760
- y, 760
- Ewl_Text: A text layout widget, 452
- ewl_text_align_apply
 - Ewl_Text, 461
- ewl_text_align_get
 - Ewl_Text, 461
- ewl_text_align_set
 - Ewl_Text, 461
- ewl_text_bg_color_apply
 - Ewl_Text, 462
- ewl_text_bg_color_get
 - Ewl_Text, 462
- ewl_text_bg_color_set
 - Ewl_Text, 462
- ewl_text_cb_child_add
 - Ewl_Text, 463
- ewl_text_cb_child_del
 - Ewl_Text, 463
- ewl_text_cb_configure
 - Ewl_Text, 463
 - lib/ewl_text.c, 1224
- ewl_text_cb_destroy
 - Ewl_Text, 463
 - lib/ewl_text.c, 1224
- ewl_text_cb_hide
 - Ewl_Text, 463
 - lib/ewl_text.c, 1224
- ewl_text_cb_mouse_down
 - Ewl_Text, 463
 - lib/ewl_text.c, 1224
- ewl_text_cb_mouse_move
 - Ewl_Text, 463
 - lib/ewl_text.c, 1224
- ewl_text_cb_mouse_up
 - Ewl_Text, 463
 - lib/ewl_text.c, 1224
- ewl_text_cb_obscure
 - Ewl_Text, 463
 - lib/ewl_text.c, 1224
- ewl_text_cb_reveal
 - Ewl_Text, 463
 - lib/ewl_text.c, 1224
- ewl_text_cb_show
 - Ewl_Text, 463
 - lib/ewl_text.c, 1224
- EWL_TEXT_CHAR_BYTE_LEN
 - lib/ewl_text.c, 1223
- ewl_text_clear
 - Ewl_Text, 463
- ewl_text_color_apply
 - Ewl_Text, 463
- ewl_text_color_get
 - Ewl_Text, 464
- ewl_text_color_set
 - Ewl_Text, 464
- Ewl_Text_Context, 761
 - align, 761
 - bg, 761
 - color, 762
 - double_underline, 762
 - Ewl_Text, 461
 - font, 762
 - font_source, 762
 - format, 762
 - glow, 762
 - outline, 762
 - ref_count, 762
 - shadow, 762
 - size, 762
 - strikethrough, 762
 - style_colors, 762
 - styles, 763
 - underline, 763
 - wrap, 763
- ewl_text_context_acquire
 - Ewl_Text, 464
- ewl_text_context_compare
 - Ewl_Text, 465
- ewl_text_context_default_create
 - Ewl_Text, 465
- ewl_text_context_dup
 - Ewl_Text, 465
- ewl_text_context_init
 - Ewl_Text, 465
- Ewl_Text_Context_Mask
 - Ewl_Enums, 153, 163
- EWL_TEXT_CONTEXT_MASK_ALIGN
 - Ewl_Enums, 163
- EWL_TEXT_CONTEXT_MASK_BG_-
COLOR
 - Ewl_Enums, 163
- EWL_TEXT_CONTEXT_MASK_COLOR
 - Ewl_Enums, 163

- EWL_TEXT_CONTEXT_MASK_-
DOUBLE_UNDERLINE_COLOR
Ewl_Enums, 164
- EWL_TEXT_CONTEXT_MASK_FONT
Ewl_Enums, 163
- EWL_TEXT_CONTEXT_MASK_-
GLOW_COLOR
Ewl_Enums, 164
- EWL_TEXT_CONTEXT_MASK_NONE
Ewl_Enums, 163
- EWL_TEXT_CONTEXT_MASK_-
OUTLINE_COLOR
Ewl_Enums, 164
- EWL_TEXT_CONTEXT_MASK_-
SHADOW_COLOR
Ewl_Enums, 164
- EWL_TEXT_CONTEXT_MASK_SIZE
Ewl_Enums, 163
- EWL_TEXT_CONTEXT_MASK_-
STRIKETHROUGH_COLOR
Ewl_Enums, 164
- EWL_TEXT_CONTEXT_MASK_STYLES
Ewl_Enums, 163
- EWL_TEXT_CONTEXT_MASK_-
UNDERLINE_COLOR
Ewl_Enums, 164
- EWL_TEXT_CONTEXT_MASK_WRAP
Ewl_Enums, 163
- ewl_text_context_new
Ewl_Text, 465
- ewl_text_context_release
Ewl_Text, 465
- ewl_text_context_shutdown
Ewl_Text, 465
- ewl_text_coord_index_map
Ewl_Text, 465
- ewl_text_cursor_position_get
Ewl_Text, 465
- ewl_text_cursor_position_line_down_get
Ewl_Text, 465
- ewl_text_cursor_position_line_up_get
Ewl_Text, 466
- ewl_text_cursor_position_set
Ewl_Text, 466
- ewl_text_double_underline_color_apply
Ewl_Text, 466
- ewl_text_double_underline_color_get
Ewl_Text, 467
- ewl_text_double_underline_color_set
Ewl_Text, 467
- ewl_text_editor.c
test_info, 815
- EWL_TEXT_EXTEND_VAL
lib/ewl_text.c, 1223
- Ewl_Text_Fmt, 764
byte_len, 764
char_len, 764
lib/ewl_text.c, 1223
tx, 764
- ewl_text_fmt_dump
Ewl_Text, 467
- ewl_text_font_apply
Ewl_Text, 467
- ewl_text_font_get
Ewl_Text, 468
- ewl_text_font_set
Ewl_Text, 468
- ewl_text_font_size_apply
Ewl_Text, 468
- ewl_text_font_size_get
Ewl_Text, 468
- ewl_text_font_size_set
Ewl_Text, 469
- ewl_text_font_source_apply
Ewl_Text, 469
- ewl_text_font_source_get
Ewl_Text, 469
- ewl_text_font_source_set
Ewl_Text, 469
- ewl_text_glow_color_apply
Ewl_Text, 470
- ewl_text_glow_color_get
Ewl_Text, 470
- ewl_text_glow_color_set
Ewl_Text, 470
- ewl_text_has_selection
Ewl_Text, 471
- ewl_text_index_geometry_map
Ewl_Text, 471
- ewl_text_init
Ewl_Text, 471
- EWL_TEXT_IS
Ewl_Text, 460
- ewl_text_length_get
Ewl_Text, 471
- ewl_text_new
Ewl_Text, 472
- ewl_text_offsets_get
Ewl_Text, 472
- ewl_text_offsets_set
Ewl_Text, 472
- ewl_text_outline_color_apply
Ewl_Text, 472
- ewl_text_outline_color_get
Ewl_Text, 473
- ewl_text_outline_color_set
Ewl_Text, 473
- ewl_text_selectable_get

- Ewl_Text, 473
- ewl_text_selectable_set
 - Ewl_Text, 474
- ewl_text_selection_get
 - Ewl_Text, 474
- EWL_TEXT_SELECTION_IS
 - Ewl_Text, 460
- ewl_text_selection_text_get
 - Ewl_Text, 474
- EWL_TEXT_SELECTION_TYPE
 - Ewl_Text, 460
- ewl_text_shadow_color_apply
 - Ewl_Text, 474
- ewl_text_shadow_color_get
 - Ewl_Text, 475
- ewl_text_shadow_color_set
 - Ewl_Text, 475
- ewl_text_strikethrough_color_apply
 - Ewl_Text, 475
- ewl_text_strikethrough_color_get
 - Ewl_Text, 476
- ewl_text_strikethrough_color_set
 - Ewl_Text, 476
- Ewl_Text_Style
 - Ewl_Enums, 153, 164
- ewl_text_style_add
 - Ewl_Text, 477
- ewl_text_style_del
 - Ewl_Text, 477
- EWL_TEXT_STYLE_DOUBLE_ -
UNDERLINE
 - Ewl_Enums, 164
- EWL_TEXT_STYLE_FAR_SHADOW
 - Ewl_Enums, 164
- EWL_TEXT_STYLE_GLOW
 - Ewl_Enums, 164
- ewl_text_style_has
 - Ewl_Text, 477
- ewl_text_style_invert
 - Ewl_Text, 477
- EWL_TEXT_STYLE_NONE
 - Ewl_Enums, 164
- EWL_TEXT_STYLE_OUTLINE
 - Ewl_Enums, 164
- EWL_TEXT_STYLE_SHADOW
 - Ewl_Enums, 164
- EWL_TEXT_STYLE_SOFT_SHADOW
 - Ewl_Enums, 164
- EWL_TEXT_STYLE_STRIKETHROUGH
 - Ewl_Enums, 164
- EWL_TEXT_STYLE_UNDERLINE
 - Ewl_Enums, 164
- ewl_text_styles_apply
 - Ewl_Text, 478
- ewl_text_styles_get
 - Ewl_Text, 478
- ewl_text_styles_set
 - Ewl_Text, 478
- ewl_text_text_append
 - Ewl_Text, 478
- ewl_text_text_delete
 - Ewl_Text, 479
- ewl_text_text_get
 - Ewl_Text, 479
- ewl_text_text_insert
 - Ewl_Text, 479
- ewl_text_text_prepend
 - Ewl_Text, 479
- ewl_text_text_set
 - Ewl_Text, 480
- EWL_TEXT_TRIGGER
 - Ewl_Text, 460
- Ewl_Text_Trigger, 765
 - areas, 765
 - char_base, 765
 - char_len, 765
 - char_pos, 766
 - Ewl_Text, 461
 - text_parent, 766
 - type, 766
 - widget, 766
- EWL_TEXT_TRIGGER_AREA
 - Ewl_Text, 460
- Ewl_Text_Trigger_Area, 767
 - deleted, 767
 - Ewl_Text, 461
 - widget, 767
- ewl_text_trigger_area_init
 - Ewl_Text, 480
- ewl_text_trigger_area_new
 - Ewl_Text, 480
- ewl_text_trigger_base_get
 - Ewl_Text, 480
- ewl_text_trigger_base_set
 - Ewl_Text, 480
- ewl_text_trigger_cb_destroy
 - Ewl_Text, 481
 - lib/ewl_text.c, 1224
- ewl_text_trigger_cb_mouse_down
 - Ewl_Text, 481
 - lib/ewl_text.c, 1224
- ewl_text_trigger_cb_mouse_in
 - Ewl_Text, 481
 - lib/ewl_text.c, 1224
- ewl_text_trigger_cb_mouse_out
 - Ewl_Text, 481
 - lib/ewl_text.c, 1224
- ewl_text_trigger_cb_mouse_up

- Ewl_Text, 481
- lib/ewl_text.c, 1224
- ewl_text_trigger_init
 - Ewl_Text, 481
- EWL_TEXT_TRIGGER_IS
 - Ewl_Text, 460
- ewl_text_trigger_length_get
 - Ewl_Text, 481
- ewl_text_trigger_length_set
 - Ewl_Text, 481
- ewl_text_trigger_new
 - Ewl_Text, 482
- ewl_text_trigger_start_pos_get
 - Ewl_Text, 482
- ewl_text_trigger_start_pos_set
 - Ewl_Text, 482
- EWL_TEXT_TRIGGER_TYPE
 - Ewl_Text, 460
- Ewl_Text_Trigger_Type
 - Ewl_Enums, 153, 164
- ewl_text_trigger_type_get
 - Ewl_Text, 482
- EWL_TEXT_TRIGGER_TYPE_NONE
 - Ewl_Enums, 164
- EWL_TEXT_TRIGGER_TYPE_-SELECTION
 - Ewl_Enums, 164
- EWL_TEXT_TRIGGER_TYPE_-TRIGGER
 - Ewl_Enums, 164
- ewl_text_triggers_configure
 - Ewl_Text, 483
- ewl_text_triggers_hide
 - Ewl_Text, 483
- ewl_text_triggers_realize
 - Ewl_Text, 483
- ewl_text_triggers_show
 - Ewl_Text, 483
- ewl_text_triggers_unrealize
 - Ewl_Text, 483
- EWL_TEXT_TYPE
 - Ewl_Text, 460
- ewl_text_underline_color_apply
 - Ewl_Text, 483
- ewl_text_underline_color_get
 - Ewl_Text, 483
- ewl_text_underline_color_set
 - Ewl_Text, 484
- Ewl_Text_Wrap
 - Ewl_Enums, 153, 164
- ewl_text_wrap_apply
 - Ewl_Text, 484
- EWL_TEXT_WRAP_CHAR
 - Ewl_Enums, 164
- ewl_text_wrap_get
 - Ewl_Text, 484
- EWL_TEXT_WRAP_NONE
 - Ewl_Enums, 164
- ewl_text_wrap_set
 - Ewl_Text, 484
- EWL_TEXT_WRAP_WORD
 - Ewl_Enums, 164
- Ewl_Theme
 - ewl_theme_data_int_get, 487
 - ewl_theme_data_int_set, 487
 - ewl_theme_data_reset, 487
 - ewl_theme_data_str_get, 487
 - ewl_theme_data_str_set, 488
 - ewl_theme_font_path_add, 488
 - ewl_theme_font_path_get, 488
 - ewl_theme_image_get, 488
 - ewl_theme_init, 489
 - EWL_THEME_KEY_NOMATCH, 487
 - ewl_theme_path_get, 489
 - ewl_theme_shutdown, 489
 - ewl_theme_theme_set, 489
 - ewl_theme_widget_init, 489
 - ewl_theme_widget_shutdown, 489
- Ewl_Theme: Methods for Accessing and Modifying Theme Data, 486
- ewl_theme_data_int_get
 - Ewl_Theme, 487
- ewl_theme_data_int_set
 - Ewl_Theme, 487
- ewl_theme_data_reset
 - Ewl_Theme, 487
- ewl_theme_data_str_get
 - Ewl_Theme, 487
- ewl_theme_data_str_set
 - Ewl_Theme, 488
- ewl_theme_font_path_add
 - Ewl_Theme, 488
- ewl_theme_font_path_get
 - Ewl_Theme, 488
- ewl_theme_image_get
 - Ewl_Theme, 488
- ewl_theme_init
 - Ewl_Theme, 489
- EWL_THEME_KEY_NOMATCH
 - Ewl_Theme, 487
- ewl_theme_path_get
 - Ewl_Theme, 489
- ewl_theme_shutdown
 - Ewl_Theme, 489
- ewl_theme_theme_set
 - Ewl_Theme, 489
- ewl_theme_widget_init
 - Ewl_Theme, 489

- ewl_theme_widget_shutdown
 - Ewl_Theme, 489
- EWL_TOOLBAR
 - Ewl_Toolbar, 491
- Ewl_Toolbar, 768
 - ewl_htoolbar_new, 491
 - EWL_TOOLBAR, 491
 - Ewl_Toolbar, 491
 - ewl_toolbar_cb_child_add, 491
 - ewl_toolbar_icon_part_hide, 491
 - ewl_toolbar_init, 491
 - EWL_TOOLBAR_IS, 491
 - ewl_toolbar_new, 492
 - ewl_toolbar_orientation_get, 492
 - ewl_toolbar_orientation_set, 492
 - EWL_TOOLBAR_TYPE, 491
 - ewl_vtoolbar_new, 492
 - hidden, 768
 - menubar, 768
- Ewl_Toolbar: A toolbar widget, 490
- ewl_toolbar_cb_child_add
 - Ewl_Toolbar, 491
- ewl_toolbar_icon_part_hide
 - Ewl_Toolbar, 491
- ewl_toolbar_init
 - Ewl_Toolbar, 491
- EWL_TOOLBAR_IS
 - Ewl_Toolbar, 491
- ewl_toolbar_new
 - Ewl_Toolbar, 492
- ewl_toolbar_orientation_get
 - Ewl_Toolbar, 492
- ewl_toolbar_orientation_set
 - Ewl_Toolbar, 492
- EWL_TOOLBAR_TYPE
 - Ewl_Toolbar, 491
- ewl_tooltip.c
 - test_info, 816
- EWL_TREE
 - Ewl_Tree, 495
- Ewl_Tree, 769
 - container, 769
 - EWL_TREE, 495
 - Ewl_Tree, 496
 - ewl_tree_cb_child_resize, 497
 - ewl_tree_cb_configure, 497
 - ewl_tree_cb_destroy, 497
 - ewl_tree_cb_header_change, 497
 - ewl_tree_cb_header_configure, 497
 - ewl_tree_cb_hscroll, 497
 - ewl_tree_cb_node_child_add, 497
 - ewl_tree_cb_node_child_del, 497
 - ewl_tree_cb_node_child_hide, 497
 - ewl_tree_cb_node_child_show, 497
 - ewl_tree_cb_node_configure, 497
 - ewl_tree_cb_node_destroy, 497
 - ewl_tree_cb_node_resize, 497
 - ewl_tree_cb_node_toggle, 497
 - ewl_tree_cb_row_hide, 497
 - ewl_tree_cb_row_select, 497
 - ewl_tree_columns_set, 497
 - ewl_tree_entry_row_add, 498
 - ewl_tree_expandable_rows_get, 498
 - ewl_tree_expandable_rows_set, 498
 - ewl_tree_headers_set, 498
 - ewl_tree_headers_visible_get, 499
 - ewl_tree_headers_visible_set, 499
 - ewl_tree_init, 499
 - EWL_TREE_IS, 495
 - ewl_tree_mode_get, 499
 - ewl_tree_mode_set, 500
 - ewl_tree_new, 500
 - EWL_TREE_NODE, 495
 - Ewl_Tree_Node, 496
 - ewl_tree_node_collapse, 500
 - ewl_tree_node_expand, 500
 - ewl_tree_node_expandable_get, 501
 - ewl_tree_node_expandable_set, 501
 - ewl_tree_node_init, 501
 - EWL_TREE_NODE_IS, 495
 - ewl_tree_node_new, 501
 - EWL_TREE_NODE_TYPE, 496
 - ewl_tree_row_add, 502
 - ewl_tree_row_column_get, 502
 - ewl_tree_row_destroy, 502
 - ewl_tree_row_expand_set, 503
 - ewl_tree_row_find, 503
 - ewl_tree_selected_clear, 503
 - ewl_tree_selected_get, 503
 - ewl_tree_text_row_add, 503
 - EWL_TREE_TYPE, 496
 - expands_visible, 769
 - header, 769
 - headers_visible, 769
 - mode, 770
 - ncols, 770
 - scrollarea, 770
 - selected, 770
- EWL_TREE2
 - Ewl_Tree2, 520
- Ewl_Tree2, 771
 - column, 772
 - columns, 772
 - content_view, 772
 - direction, 772
 - EWL_TREE2, 520
 - Ewl_Tree2, 521
 - ewl_tree2_cb_column_sort, 522

- ewl_tree2_cb_configure, 522
- ewl_tree2_cb_destroy, 522
- ewl_tree2_cb_node_child_add, 522
- ewl_tree2_cb_node_child_del, 522
- ewl_tree2_cb_node_child_hide, 522
- ewl_tree2_cb_node_child_show, 522
- ewl_tree2_cb_node_configure, 522
- ewl_tree2_cb_node_realize, 522
- ewl_tree2_cb_node_resize, 522
- ewl_tree2_cb_node_toggle, 522
- ewl_tree2_column_count_get, 522
- ewl_tree2_column_count_set, 522
- ewl_tree2_content_view_get, 523
- ewl_tree2_content_view_set, 523
- ewl_tree2_content_widget_get, 523
- ewl_tree2_fixed_rows_get, 523
- ewl_tree2_fixed_rows_set, 523
- ewl_tree2_headers_visible_get, 524
- ewl_tree2_headers_visible_set, 524
- ewl_tree2_init, 524
- EWL_TREE2_IS, 520
- ewl_tree2_new, 524
- EWL_TREE2_NODE, 520
- Ewl_Tree2_Node, 521
- ewl_tree2_node_collapse, 525
- ewl_tree2_node_expand, 525
- ewl_tree2_node_expandable_get, 525
- ewl_tree2_node_expandable_set, 525
- ewl_tree2_node_expanded_is, 525
- ewl_tree2_node_init, 525
- EWL_TREE2_NODE_IS, 521
- ewl_tree2_node_new, 525
- EWL_TREE2_NODE_TYPE, 521
- ewl_tree2_row_collapse, 525
- ewl_tree2_row_expand, 525
- ewl_tree2_row_expanded_is, 525
- ewl_tree2_selection_type_get, 526
- ewl_tree2_selection_type_set, 526
- EWL_TREE2_TYPE, 521
- expansions, 772
- fixed, 772
- header, 772
- headers_visible, 772
- mvc, 772
- rowcache, 772
- rows, 772
- sort, 772
- type, 773
- Ewl_Tree2: A Widget for List or Tree Layout, 505
- ewl_tree2_cb_column_sort
 - Ewl_Tree2, 522
 - lib/ewl_tree2.c, 1252
- ewl_tree2_cb_configure
 - Ewl_Tree2, 522
 - lib/ewl_tree2.c, 1252
- ewl_tree2_cb_destroy
 - Ewl_Tree2, 522
 - lib/ewl_tree2.c, 1252
- ewl_tree2_cb_node_child_add
 - Ewl_Tree2, 522
 - lib/ewl_tree2.c, 1252
- ewl_tree2_cb_node_child_del
 - Ewl_Tree2, 522
- ewl_tree2_cb_node_child_hide
 - Ewl_Tree2, 522
- ewl_tree2_cb_node_child_show
 - Ewl_Tree2, 522
 - lib/ewl_tree2.c, 1252
- ewl_tree2_cb_node_configure
 - Ewl_Tree2, 522
 - lib/ewl_tree2.c, 1252
- ewl_tree2_cb_node_realize
 - Ewl_Tree2, 522
- ewl_tree2_cb_node_resize
 - Ewl_Tree2, 522
- ewl_tree2_cb_node_toggle
 - Ewl_Tree2, 522
 - lib/ewl_tree2.c, 1252
- ewl_tree2_column_count_get
 - Ewl_Tree2, 522
- ewl_tree2_column_count_set
 - Ewl_Tree2, 522
- ewl_tree2_content_view_get
 - Ewl_Tree2, 523
- ewl_tree2_content_view_set
 - Ewl_Tree2, 523
- ewl_tree2_content_widget_get
 - Ewl_Tree2, 523
- ewl_tree2_fixed_rows_get
 - Ewl_Tree2, 523
- ewl_tree2_fixed_rows_set
 - Ewl_Tree2, 523
- ewl_tree2_headers_visible_get
 - Ewl_Tree2, 524
- ewl_tree2_headers_visible_set
 - Ewl_Tree2, 524
- ewl_tree2_init
 - Ewl_Tree2, 524
- EWL_TREE2_IS
 - Ewl_Tree2, 520
- ewl_tree2_new
 - Ewl_Tree2, 524
- EWL_TREE2_NODE
 - Ewl_Tree2, 520
- Ewl_Tree2_Node, 774
 - built_children, 774
 - Ewl_Tree2, 521

- expanded, 774
- handle, 775
- mvc, 775
- row, 775
- row_num, 775
- tree, 775
- ewl_tree2_node_collapse
 - Ewl_Tree2, 525
- ewl_tree2_node_expand
 - Ewl_Tree2, 525
- ewl_tree2_node_expandable_get
 - Ewl_Tree2, 525
- ewl_tree2_node_expandable_set
 - Ewl_Tree2, 525
- ewl_tree2_node_expanded_is
 - Ewl_Tree2, 525
- ewl_tree2_node_init
 - Ewl_Tree2, 525
- EWL_TREE2_NODE_IS
 - Ewl_Tree2, 521
- ewl_tree2_node_new
 - Ewl_Tree2, 525
- EWL_TREE2_NODE_TYPE
 - Ewl_Tree2, 521
- ewl_tree2_row_collapse
 - Ewl_Tree2, 525
- ewl_tree2_row_expand
 - Ewl_Tree2, 525
- ewl_tree2_row_expanded_is
 - Ewl_Tree2, 525
- ewl_tree2_selection_type_get
 - Ewl_Tree2, 526
- ewl_tree2_selection_type_set
 - Ewl_Tree2, 526
- EWL_TREE2_TYPE
 - Ewl_Tree2, 521
- EWL_TREE2_VIEW
 - Ewl_Tree2_View, 527
- Ewl_Tree2_View, 776
 - box, 776
 - EWL_TREE2_VIEW, 527
 - Ewl_Tree2_View, 528
 - ewl_tree2_view_init, 528
 - EWL_TREE2_VIEW_IS, 527
 - ewl_tree2_view_tree2_get, 528
 - ewl_tree2_view_tree2_set, 528
 - EWL_TREE2_VIEW_TYPE, 527
 - parent, 776
- Ewl_Tree2_View: A tree view, 527
- ewl_tree2_view_init
 - Ewl_Tree2_View, 528
- EWL_TREE2_VIEW_IS
 - Ewl_Tree2_View, 527
- EWL_TREE2_VIEW_PLAIN
 - Ewl_Tree2_View_Plain, 529
- Ewl_Tree2_View_Plain, 778
 - EWL_TREE2_VIEW_PLAIN, 529
 - Ewl_Tree2_View_Plain, 530
 - ewl_tree2_view_plain_get, 530
 - ewl_tree2_view_plain_init, 530
 - EWL_TREE2_VIEW_PLAIN_IS, 529
 - ewl_tree2_view_plain_new, 530
 - EWL_TREE2_VIEW_PLAIN_TYPE, 529
 - view, 778
- Ewl_Tree2_View_Plain: The plain tree view, 529
 - ewl_tree2_view_plain_get
 - Ewl_Tree2_View_Plain, 530
 - ewl_tree2_view_plain_init
 - Ewl_Tree2_View_Plain, 530
 - EWL_TREE2_VIEW_PLAIN_IS
 - Ewl_Tree2_View_Plain, 529
 - ewl_tree2_view_plain_new
 - Ewl_Tree2_View_Plain, 530
 - EWL_TREE2_VIEW_PLAIN_TYPE
 - Ewl_Tree2_View_Plain, 529
 - EWL_TREE2_VIEW_SCROLLED
 - Ewl_Tree2_View_Scrolled, 532
 - Ewl_Tree2_View_Scrolled, 779
 - box, 779
 - EWL_TREE2_VIEW_SCROLLED, 532
 - Ewl_Tree2_View_Scrolled, 532
 - ewl_tree2_view_scrolled_get, 532
 - ewl_tree2_view_scrolled_init, 532
 - EWL_TREE2_VIEW_SCROLLED_IS, 532
 - ewl_tree2_view_scrolled_new, 532
 - ewl_tree2_view_scrolled_scroll_ -
 - headers_get, 533
 - ewl_tree2_view_scrolled_scroll_ -
 - headers_set, 533
 - EWL_TREE2_VIEW_SCROLLED_ -
 - TYPE, 532
 - scroll, 779
 - scroll_header, 779
 - scroll_visible, 779
 - Ewl_Tree2_View_Scrolled The scrolled tree view, 531
 - ewl_tree2_view_scrolled_get
 - Ewl_Tree2_View_Scrolled, 532
 - ewl_tree2_view_scrolled_init
 - Ewl_Tree2_View_Scrolled, 532
 - EWL_TREE2_VIEW_SCROLLED_IS
 - Ewl_Tree2_View_Scrolled, 532
 - ewl_tree2_view_scrolled_new
 - Ewl_Tree2_View_Scrolled, 532
 - ewl_tree2_view_scrolled_scroll_headers_get

- Ewl_Tree2_View_Scrolled, 533
- ewl_tree2_view_scrolled_scroll_headers_set
 - Ewl_Tree2_View_Scrolled, 533
- EWL_TREE2_VIEW_SCROLLED_TYPE
 - Ewl_Tree2_View_Scrolled, 532
- ewl_tree2_view_tree2_get
 - Ewl_Tree2_View, 528
- ewl_tree2_view_tree2_set
 - Ewl_Tree2_View, 528
- EWL_TREE2_VIEW_TYPE
 - Ewl_Tree2_View, 527
- Ewl_Tree: A Widget for List or Tree Layout, 493
- ewl_tree_cb_child_resize
 - Ewl_Tree, 497
- ewl_tree_cb_configure
 - Ewl_Tree, 497
 - lib/ewl_tree.c, 1245
- ewl_tree_cb_destroy
 - Ewl_Tree, 497
 - lib/ewl_tree.c, 1245
- ewl_tree_cb_header_change
 - Ewl_Tree, 497
 - lib/ewl_tree.c, 1245
- ewl_tree_cb_header_configure
 - Ewl_Tree, 497
 - lib/ewl_tree.c, 1245
- ewl_tree_cb_hscroll
 - Ewl_Tree, 497
 - lib/ewl_tree.c, 1245
- ewl_tree_cb_node_child_add
 - Ewl_Tree, 497
 - lib/ewl_tree.c, 1245
- ewl_tree_cb_node_child_del
 - Ewl_Tree, 497
- ewl_tree_cb_node_child_hide
 - Ewl_Tree, 497
- ewl_tree_cb_node_child_show
 - Ewl_Tree, 497
 - lib/ewl_tree.c, 1245
- ewl_tree_cb_node_configure
 - Ewl_Tree, 497
 - lib/ewl_tree.c, 1245
- ewl_tree_cb_node_destroy
 - Ewl_Tree, 497
 - lib/ewl_tree.c, 1245
- ewl_tree_cb_node_resize
 - Ewl_Tree, 497
- ewl_tree_cb_node_toggle
 - Ewl_Tree, 497
 - lib/ewl_tree.c, 1245
- ewl_tree_cb_row_hide
 - Ewl_Tree, 497
 - lib/ewl_tree.c, 1245
- ewl_tree_cb_row_select
 - Ewl_Tree, 497
 - lib/ewl_tree.c, 1245
- ewl_tree_columns_set
 - Ewl_Tree, 497
- ewl_tree_entry_row_add
 - Ewl_Tree, 498
- ewl_tree_expandable_rows_get
 - Ewl_Tree, 498
- ewl_tree_expandable_rows_set
 - Ewl_Tree, 498
- ewl_tree_headers_set
 - Ewl_Tree, 498
- ewl_tree_headers_visible_get
 - Ewl_Tree, 499
- ewl_tree_headers_visible_set
 - Ewl_Tree, 499
- ewl_tree_init
 - Ewl_Tree, 499
- EWL_TREE_IS
 - Ewl_Tree, 495
- ewl_tree_mode_get
 - Ewl_Tree, 499
- ewl_tree_mode_set
 - Ewl_Tree, 500
- ewl_tree_new
 - Ewl_Tree, 500
- EWL_TREE_NODE
 - Ewl_Tree, 495
- Ewl_Tree_Node, 781
 - container, 781
 - Ewl_Tree, 496
 - expanded, 781
 - handle, 781
 - row, 781
 - tree, 782
- ewl_tree_node_collapse
 - Ewl_Tree, 500
- EWL_TREE_NODE_COLLAPSED
 - Ewl_Enums, 165
- ewl_tree_node_expand
 - Ewl_Tree, 500
- ewl_tree_node_expandable_get
 - Ewl_Tree, 501
- ewl_tree_node_expandable_set
 - Ewl_Tree, 501
- EWL_TREE_NODE_EXPANDED
 - Ewl_Enums, 165
- Ewl_Tree_Node_Flags
 - Ewl_Enums, 153, 164
- ewl_tree_node_init
 - Ewl_Tree, 501
- EWL_TREE_NODE_IS
 - Ewl_Tree, 495

- ewl_tree_node_new
 - Ewl_Tree, 501
- EWL_TREE_NODE_NOEXPAND
 - Ewl_Enums, 165
- EWL_TREE_NODE_TYPE
 - Ewl_Tree, 496
- ewl_tree_row_add
 - Ewl_Tree, 502
- ewl_tree_row_column_get
 - Ewl_Tree, 502
- ewl_tree_row_destroy
 - Ewl_Tree, 502
- ewl_tree_row_expand_set
 - Ewl_Tree, 503
- ewl_tree_row_find
 - Ewl_Tree, 503
- ewl_tree_row_remove
 - lib/ewl_tree.c, 1245
- ewl_tree_selected_clear
 - Ewl_Tree, 503
- ewl_tree_selected_get
 - Ewl_Tree, 503
- Ewl_Tree_Selection_Type
 - Ewl_Enums, 153, 165
- EWL_TREE_SELECTION_TYPE_CELL
 - Ewl_Enums, 165
- EWL_TREE_SELECTION_TYPE_ROW
 - Ewl_Enums, 165
- ewl_tree_text_row_add
 - Ewl_Tree, 503
- EWL_TREE_TYPE
 - Ewl_Tree, 496
- Ewl_Unit_Test, 783
 - Ewl_Test.h, 800
 - failure_reason, 783
 - func, 783
 - name, 783
 - status, 783
- ewl_vbox_new
 - Ewl_Box, 34
- ewl_vfreebox_new
 - Ewl_Freebox, 217
- EWL_VIEW
 - Ewl_View, 535
- Ewl_View, 784
 - EWL_VIEW, 535
 - Ewl_View, 535
 - ewl_view_clone, 536
 - Ewl_View_Expansion_View_Fetch, 535
 - EWL_VIEW_EXPANSION_VIEW_-
FETCH_GET, 535
 - ewl_view_expansion_view_fetch_get,
536
 - ewl_view_expansion_view_fetch_set,
536
 - Ewl_View_Header_Fetch, 535
 - ewl_view_header_fetch_get, 536
 - ewl_view_header_fetch_set, 537
 - EWL_VIEW_HEADER_GET, 535
 - ewl_view_init, 537
 - ewl_view_new, 537
 - EWL_VIEW_WIDGET_FETCH, 535
 - Ewl_View_Widget_Fetch, 535
 - ewl_view_widget_fetch_get, 537
 - ewl_view_widget_fetch_set, 538
 - ewl_vmenubar_new
 - Ewl_Menubar, 310
 - ewl_vpaned_new
 - Ewl_Paned, 378
 - ewl_vscrollbar_new
 - Ewl_Scrollbar, 408
 - ewl_vseeker_new

- Ewl_Seeker, 418
- ewl_vseparator_new
 - Ewl_Separator, 421
- ewl_vtoolbar_new
 - Ewl_Toolbar, 492
- EWL_WIDGET
 - Ewl_Widget, 545
- Ewl_Widget, 785
 - appearance, 785
 - attach, 785
 - callbacks, 785
 - Ewl_Attach_List, 546
 - Ewl_Callback_Chain, 546
 - Ewl_Color_Set, 546
 - EWL_PAIR, 545
 - Ewl_Pair, 546
 - Ewl_Pair_List, 546
 - EWL_WIDGET, 545
 - Ewl_Widget, 546
 - ewl_widget_appearance_get, 546
 - ewl_widget_appearance_part_text_get, 546
 - ewl_widget_appearance_part_text_set, 547
 - ewl_widget_appearance_path_copy, 547
 - ewl_widget_appearance_path_get, 547
 - ewl_widget_appearance_path_size_get, 547
 - ewl_widget_appearance_set, 548
 - ewl_widget_appearance_text_get, 548
 - ewl_widget_appearance_text_set, 548
 - ewl_widget_cb_configure, 548
 - ewl_widget_cb_disable, 549
 - ewl_widget_cb_enable, 549
 - ewl_widget_cb_focus_in, 549
 - ewl_widget_cb_focus_out, 549
 - ewl_widget_cb_hide, 549
 - ewl_widget_cb_mouse_down, 549
 - ewl_widget_cb_mouse_in, 549
 - ewl_widget_cb_mouse_move, 549
 - ewl_widget_cb_mouse_out, 549
 - ewl_widget_cb_mouse_up, 549
 - ewl_widget_cb_obscure, 549
 - ewl_widget_cb_realize, 549
 - ewl_widget_cb_reparent, 549
 - ewl_widget_cb_reveal, 549
 - ewl_widget_cb_show, 549
 - ewl_widget_cb_unrealize, 549
 - ewl_widget_clipped_is, 549
 - ewl_widget_clipped_set, 550
 - ewl_widget_color_get, 550
 - ewl_widget_color_set, 550
 - ewl_widget_configure, 550
 - ewl_widget_data_del, 551
 - ewl_widget_data_get, 551
 - ewl_widget_data_set, 551
 - ewl_widget_destroy, 552
 - ewl_widget_disable, 552
 - ewl_widget_dnd_reset, 552
 - Ewl_Widget_Drag, 546
 - ewl_widget_drag_candidate_get, 552
 - ewl_widget_draggable_set, 552
 - ewl_widget_enable, 553
 - ewl_widget_focus_send, 553
 - ewl_widget_focusable_get, 553
 - ewl_widget_focusable_set, 553
 - ewl_widget_focused_get, 554
 - ewl_widget_free, 554
 - ewl_widget_hide, 554
 - ewl_widget_ignore_focus_change_get, 554
 - ewl_widget_ignore_focus_change_set, 554
 - ewl_widget_inherit, 555
 - ewl_widget_init, 555
 - ewl_widget_internal_is, 555
 - ewl_widget_internal_set, 555
 - EWL_WIDGET_IS, 545
 - ewl_widget_layer_priority_get, 555
 - ewl_widget_layer_priority_set, 555
 - ewl_widget_layer_top_get, 556
 - ewl_widget_layer_top_set, 556
 - ewl_widget_name_find, 556
 - ewl_widget_name_get, 556
 - ewl_widget_name_set, 557
 - ewl_widget_obscure, 557
 - ewl_widget_onscreen_is, 557
 - ewl_widget_parent_get, 557
 - ewl_widget_parent_of, 558
 - ewl_widget_parent_set, 558
 - ewl_widget_print, 558
 - ewl_widget_print_verbose, 558
 - ewl_widget_realize, 559
 - ewl_widget_reparent, 559
 - ewl_widget_reveal, 559
 - ewl_widget_show, 559
 - ewl_widget_state_set, 560
 - ewl_widget_tab_order_append, 560
 - ewl_widget_tab_order_insert, 560
 - ewl_widget_tab_order_insert_after, 561
 - ewl_widget_tab_order_insert_before, 561
 - ewl_widget_tab_order_prepend, 561
 - ewl_widget_tab_order_remove, 561
 - ewl_widget_tree_print, 562
 - EWL_WIDGET_TYPE, 545
 - ewl_widget_type_is, 562
 - ewl_widget_unrealize, 562

- fx_clip_box, 786
- inheritance, 786
- layer, 786
- object, 786
- parent, 786
- smart_object, 786
- theme, 786
- theme_group, 786
- theme_object, 786
- theme_path, 786
- theme_state, 786
- theme_text, 787
- toplayered, 787
- Ewl_Widget: The Parent Widget Class Common to All Widgets, 539
- ewl_widget_appearance_get
 - Ewl_Widget, 546
- ewl_widget_appearance_part_text_get
 - Ewl_Widget, 546
- ewl_widget_appearance_part_text_set
 - Ewl_Widget, 547
- ewl_widget_appearance_path_copy
 - Ewl_Widget, 547
- ewl_widget_appearance_path_get
 - Ewl_Widget, 547
- ewl_widget_appearance_path_size_get
 - Ewl_Widget, 547
- ewl_widget_appearance_set
 - Ewl_Widget, 548
- ewl_widget_appearance_text_get
 - Ewl_Widget, 548
- ewl_widget_appearance_text_set
 - Ewl_Widget, 548
- ewl_widget_cb_configure
 - Ewl_Widget, 548
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_disable
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_drag_down
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_drag_move
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_drag_up
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_enable
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_focus_in
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_focus_out
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_hide
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_mouse_down
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_mouse_in
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_mouse_move
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_mouse_out
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_mouse_up
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_obscure
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_realize
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_reparent
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_reveal
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_show
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_cb_unrealize
 - Ewl_Widget, 549
 - lib/ewl_widget.c, 1275
- ewl_widget_clipped_is
 - Ewl_Widget, 549
- ewl_widget_clipped_set
 - Ewl_Widget, 550
- ewl_widget_color_get
 - Ewl_Widget, 550
- ewl_widget_color_set
 - Ewl_Widget, 550
- ewl_widget_configure
 - Ewl_Widget, 550
- ewl_widget_data_del
 - Ewl_Widget, 551
- ewl_widget_data_get
 - Ewl_Widget, 551
- ewl_widget_data_set
 - Ewl_Widget, 551
- ewl_widget_destroy
 - Ewl_Widget, 552

- ewl_widget_disable
 - Ewl_Widget, 552
- ewl_widget_dnd_reset
 - Ewl_Widget, 552
- Ewl_Widget_Drag
 - Ewl_Widget, 546
- ewl_widget_drag_candidate_get
 - Ewl_Widget, 552
- ewl_widget_draggable_set
 - Ewl_Widget, 552
- ewl_widget_enable
 - Ewl_Widget, 553
- ewl_widget_focus_send
 - Ewl_Widget, 553
- ewl_widget_focusable_get
 - Ewl_Widget, 553
- ewl_widget_focusable_set
 - Ewl_Widget, 553
- ewl_widget_focused_get
 - Ewl_Widget, 554
- ewl_widget_free
 - Ewl_Widget, 554
- ewl_widget_hide
 - Ewl_Widget, 554
- ewl_widget_ignore_focus_change_get
 - Ewl_Widget, 554
- ewl_widget_ignore_focus_change_set
 - Ewl_Widget, 554
- ewl_widget_inherit
 - Ewl_Widget, 555
- ewl_widget_init
 - Ewl_Widget, 555
- ewl_widget_internal_is
 - Ewl_Widget, 555
- ewl_widget_internal_set
 - Ewl_Widget, 555
- EWL_WIDGET_IS
 - Ewl_Widget, 545
- ewl_widget_layer_priority_get
 - Ewl_Widget, 555
- ewl_widget_layer_priority_set
 - Ewl_Widget, 555
- ewl_widget_layer_top_get
 - Ewl_Widget, 556
- ewl_widget_layer_top_set
 - Ewl_Widget, 556
- ewl_widget_name_find
 - Ewl_Widget, 556
- ewl_widget_name_get
 - Ewl_Widget, 556
- ewl_widget_name_set
 - Ewl_Widget, 557
- ewl_widget_new
 - lib/ewl_widget.c, 1275
- ewl_widget_obscure
 - Ewl_Widget, 557
- ewl_widget_onscreen_is
 - Ewl_Widget, 557
- ewl_widget_parent_get
 - Ewl_Widget, 557
- ewl_widget_parent_of
 - Ewl_Widget, 558
- ewl_widget_parent_set
 - Ewl_Widget, 558
- ewl_widget_print
 - Ewl_Widget, 558
- ewl_widget_print_verbose
 - Ewl_Widget, 558
- ewl_widget_realize
 - Ewl_Widget, 559
- ewl_widget_reparent
 - Ewl_Widget, 559
- ewl_widget_reveal
 - Ewl_Widget, 559
- ewl_widget_show
 - Ewl_Widget, 559
- ewl_widget_state_set
 - Ewl_Widget, 560
- ewl_widget_tab_order_append
 - Ewl_Widget, 560
- ewl_widget_tab_order_insert
 - Ewl_Widget, 560
- ewl_widget_tab_order_insert_after
 - Ewl_Widget, 561
- ewl_widget_tab_order_insert_before
 - Ewl_Widget, 561
- ewl_widget_tab_order_prepend
 - Ewl_Widget, 561
- ewl_widget_tab_order_remove
 - Ewl_Widget, 561
- ewl_widget_tree_print
 - Ewl_Widget, 562
- EWL_WIDGET_TYPE
 - Ewl_Widget, 545
- ewl_widget_type_is
 - Ewl_Widget, 562
- ewl_widget_unrealize
 - Ewl_Widget, 562
- EWL_WINDOW
 - Ewl_Window, 566
- Ewl_Window, 788
 - classname, 789
 - embed, 789
 - ewl, 789
 - EWL_CALLBACK_DELETE_-
WINDOW, 575
 - EWL_CALLBACK_EXPOSE, 575
 - EWL_WINDOW, 566

- Ewl_Window, 566
- ewl_window_attention_demand, 566
- ewl_window_borderless_set, 566
- ewl_window_cb_configure, 566
- ewl_window_cb_destroy, 567
- ewl_window_cb_expose, 567
- ewl_window_cb_hide, 567
- ewl_window_cb_postrealize, 567
- ewl_window_cb_realize, 567
- ewl_window_cb_realize_transient, 567
- ewl_window_cb_show, 567
- ewl_window_cb_unrealize, 567
- ewl_window_class_get, 567
- ewl_window_class_set, 567
- ewl_window_dialog_get, 568
- ewl_window_dialog_set, 568
- ewl_window_fullscreen_get, 568
- ewl_window_fullscreen_set, 568
- ewl_window_init, 568
- EWL_WINDOW_IS, 566
- ewl_window_keyboard_grab_get, 569
- ewl_window_keyboard_grab_set, 569
- ewl_window_lower, 569
- ewl_window_modal_get, 569
- ewl_window_modal_set, 570
- ewl_window_move, 570
- ewl_window_name_get, 570
- ewl_window_name_set, 570
- ewl_window_new, 571
- ewl_window_override_get, 571
- ewl_window_override_set, 571
- ewl_window_pointer_grab_get, 571
- ewl_window_pointer_grab_set, 572
- ewl_window_raise, 572
- ewl_window_selection_text_set, 572
- ewl_window_skip_pager_get, 572
- ewl_window_skip_pager_set, 573
- ewl_window_skip_taskbar_get, 573
- ewl_window_skip_taskbar_set, 573
- ewl_window_title_get, 573
- ewl_window_title_set, 574
- ewl_window_transient_for, 574
- ewl_window_transient_for_foreign, 574
- EWL_WINDOW_TYPE, 566
- ewl_window_window_find, 574
- flags, 789
- foreign, 789
- name, 789
- old_flags, 789
- title, 789
- transient, 789
- window, 789
- ewl_window.c
 - ewl_window_cb_configure, 1287
 - ewl_window_cb_destroy, 1287
 - ewl_window_cb_expose, 1287
 - ewl_window_cb_hide, 1287
 - ewl_window_cb_postrealize, 1287
 - ewl_window_cb_realize, 1287
 - ewl_window_cb_realize_transient, 1287
 - ewl_window_cb_show, 1287
 - ewl_window_cb_unrealize, 1287
 - ewl_window_list, 1287
- Ewl_Window: A Container for Displaying in a New Window, 563
- ewl_window_attention_demand
 - Ewl_Window, 566
- EWL_WINDOW_BORDERLESS
 - Ewl_Enums, 165
- ewl_window_borderless_set
 - Ewl_Window, 566
- ewl_window_cb_configure
 - Ewl_Window, 566
 - ewl_window.c, 1287
- ewl_window_cb_destroy
 - Ewl_Window, 567
 - ewl_window.c, 1287
- ewl_window_cb_expose
 - Ewl_Window, 567
 - ewl_window.c, 1287
- ewl_window_cb_hide
 - Ewl_Window, 567
 - ewl_window.c, 1287
- ewl_window_cb_postrealize
 - Ewl_Window, 567
 - ewl_window.c, 1287
- ewl_window_cb_realize
 - Ewl_Window, 567
 - ewl_window.c, 1287
- ewl_window_cb_realize_transient
 - Ewl_Window, 567
 - ewl_window.c, 1287
- ewl_window_cb_show
 - Ewl_Window, 567
 - ewl_window.c, 1287
- ewl_window_cb_unrealize
 - Ewl_Window, 567
 - ewl_window.c, 1287
- ewl_window_class_get
 - Ewl_Window, 567
- ewl_window_class_set
 - Ewl_Window, 567
- EWL_WINDOW_DEMANDS_ATTENTION
 - Ewl_Enums, 165
- EWL_WINDOW_DIALOG
 - Ewl_Enums, 165
- ewl_window_dialog_get
 - Ewl_Window, 568

- ewl_window_dialog_set
 - Ewl_Window, 568
- Ewl_Window_Flags
 - Ewl_Enums, 154, 165
- EWL_WINDOW_FULLSCREEN
 - Ewl_Enums, 165
- ewl_window_fullscreen_get
 - Ewl_Window, 568
- ewl_window_fullscreen_set
 - Ewl_Window, 568
- EWL_WINDOW_GRAB_KEYBOARD
 - Ewl_Enums, 165
- EWL_WINDOW_GRAB_POINTER
 - Ewl_Enums, 165
- ewl_window_init
 - Ewl_Window, 568
- EWL_WINDOW_IS
 - Ewl_Window, 566
- ewl_window_keyboard_grab_get
 - Ewl_Window, 569
- ewl_window_keyboard_grab_set
 - Ewl_Window, 569
- ewl_window_list
 - ewl_misc.c, 1101
 - ewl_window.c, 1287
- ewl_window_lower
 - Ewl_Window, 569
- EWL_WINDOW_MODAL
 - Ewl_Enums, 165
- ewl_window_modal_get
 - Ewl_Window, 569
- ewl_window_modal_set
 - Ewl_Window, 570
- ewl_window_move
 - Ewl_Window, 570
- ewl_window_name_get
 - Ewl_Window, 570
- ewl_window_name_set
 - Ewl_Window, 570
- ewl_window_new
 - Ewl_Window, 571
- EWL_WINDOW_OVERRIDE
 - Ewl_Enums, 165
- ewl_window_override_get
 - Ewl_Window, 571
- ewl_window_override_set
 - Ewl_Window, 571
- ewl_window_pointer_grab_get
 - Ewl_Window, 571
- ewl_window_pointer_grab_set
 - Ewl_Window, 572
- ewl_window_raise
 - Ewl_Window, 572
- ewl_window_selection_text_set
 - Ewl_Window, 572
- EWL_WINDOW_SKIP_PAGER
 - Ewl_Enums, 165
- ewl_window_skip_pager_get
 - Ewl_Window, 572
- ewl_window_skip_pager_set
 - Ewl_Window, 573
- EWL_WINDOW_SKIP_TASKBAR
 - Ewl_Enums, 165
- ewl_window_skip_taskbar_get
 - Ewl_Window, 573
- ewl_window_skip_taskbar_set
 - Ewl_Window, 573
- ewl_window_title_get
 - Ewl_Window, 573
- ewl_window_title_set
 - Ewl_Window, 574
- EWL_WINDOW_TRANSIENT
 - Ewl_Enums, 165
- ewl_window_transient_for
 - Ewl_Window, 574
- ewl_window_transient_for_foreign
 - Ewl_Window, 574
- EWL_WINDOW_TRANSIENT_FOREIGN
 - Ewl_Enums, 165
- EWL_WINDOW_TYPE
 - Ewl_Window, 566
- EWL_WINDOW_USER_CONFIGURE
 - Ewl_Enums, 165
- ewl_window_window_find
 - Ewl_Window, 574
- expandable
 - Ewl_Expansion, 655
 - Tree2_Test_Row_Data, 794
- expanded
 - Ewl_Tree2_Node, 774
 - Ewl_Tree_Node, 781
- expands_visible
 - Ewl_Tree, 769
- expansion
 - Ewl_Model, 703
 - Ewl_View, 784
- expansions
 - Ewl_Tree2, 772
- extended
 - Ewl_Icon, 684
- f1_align
 - Ewl_Box_Orientation, 588
- f3_align
 - Ewl_Box_Orientation, 588
- f_policy
 - Ewl_Box_Orientation, 588
- failure_reason

- Ewl_Unit_Test, 783
- favorites_box
 - Ewl_Filepicker, 668
- fetch
 - Ewl_Model, 703
 - Ewl_View, 784
- file_entry
 - Ewl_Filepicker, 669
- file_list
 - Ewl_Filepicker, 669
- file_list_box
 - Ewl_Filepicker, 669
- file_name_get
 - Ewl_Filelist, 659
- filename
 - Ewl_Test, 755
- files
 - Ewl_Filelist_Tree_Data, 667
- fill_ask
 - Ewl_Box_Orientation, 588
- fill_percentage
 - Ewl_Scrollbar, 731
- fill_set
 - Ewl_Box_Orientation, 588
- filter
 - Ewl_Filelist, 659
 - Ewl_Filepicker_Filter, 670
- filter_change
 - Ewl_Filelist, 659
- filters
 - Ewl_Filepicker, 669
- fit_to_follow
 - Ewl_Popup, 722
- fixed
 - Ewl_Tree2, 772
- flags
 - Ewl_Object, 710
 - Ewl_Window, 789
- floater
 - Ewl_Highlight, 680
- focus
 - Ewl_Embed, 631
- focused
 - Ewl_Embed, 631
- follow
 - Ewl_Popup, 722
- follows
 - Ewl_Floater, 671
- font
 - Ewl_Text_Context, 762
- font_source
 - Ewl_Text_Context, 762
- foreign
 - Ewl_Window, 789
- format
 - Ewl_Event_Dnd_Data_Received, 640
 - Ewl_Text_Context, 762
- formatting
 - Ewl_Text, 759
- fp
 - Ewl_Filedialog, 657
- FREE
 - Ewl_Macros, 292
- freebox
 - Ewl_Filelist_Icon, 664
- Freebox_Test, 790
 - bin/tests/freebox/ewl_freebox.c, 1023
 - compare, 790
 - name, 790
 - type, 790
- func
 - Control, 578
 - Ewl_Callback, 593
 - Ewl_Test, 755
 - Ewl_Unit_Test, 783
- functions
 - Ewl_Engine, 633
- fx_clip_box
 - Ewl_Widget, 786
- g
 - Ewl_Color_Set, 600
 - Ewl_Colorpicker, 604
- G_CALC
 - lib/ewl_histogram.c, 1037
- gc_reap
 - Ewl_Config_Cache, 615
- glow
 - Ewl_Text_Context, 762
- graph
 - Ewl_Histogram, 681
- grid
 - Ewl_Calendar, 592
 - Ewl_Table, 754
- h
 - Ewl_Colorpicker, 604
 - Ewl_Event_Window_Expose, 654
 - Ewl_Filelist, 659
 - Ewl_Image, 687
 - Ewl_Object, 710
 - Ewl_Spectrum, 744
- handle
 - Ewl_Cursor, 622
 - Ewl_Engine, 633
 - Ewl_Event_Dnd_Data_Request, 641
 - Ewl_IO_Manager_Plugin, 690
 - Ewl_Test, 755

- Ewl_Tree2_Node, 775
- Ewl_Tree_Node, 781
- has_alpha
 - Ewl_Colorpicker, 604
- hbox
 - Ewl_Filelist_Column, 661
- header
 - Ewl_Combo, 610
 - Ewl_Model, 703
 - Ewl_Row, 729
 - Ewl_Tree, 769
 - Ewl_Tree2, 772
- header_fetch
 - Ewl_View, 784
- headers_visible
 - Ewl_Tree, 769
 - Ewl_Tree2, 772
- hflag
 - Ewl_Scrollpane, 732
- HIDDEN
 - Ewl_Object, 353
- hidden
 - Ewl_Icon, 684
 - Ewl_Toolbar, 768
- highlight
 - Ewl_Selection, 736
- homogeneous
 - Ewl_Box, 586
- homogeneous_h
 - Ewl_Grid, 676
 - Ewl_Table, 754
- homogeneous_v
 - Ewl_Grid, 676
 - Ewl_Table, 754
- hooks
 - Ewl_Engine_Info, 634
- horizontal
 - Ewl_Spectrum, 744
- hscrollbar
 - Ewl_Scrollpane, 732
- hsv
 - Ewl_Colorpicker, 604
 - Ewl_Spectrum, 744
- icons
 - bin/tests/icon_theme/ewl_icon_theme.c, 1049
- id
 - Ewl_Callback, 593
- IF_FREE
 - Ewl_Macros, 292
- IF_FREE_HASH
 - Ewl_Macros, 292
- IF_FREE_LIST
 - Ewl_Macros, 293
- IF_RELEASE
 - Ewl_Macros, 293
- image
 - Ewl_Histogram, 681
 - Ewl_Icon, 684
 - Ewl_Image, 687
 - Ewl_Image_Thumbnail, 689
 - List_Test_Row_Data, 792
 - Tree2_Test_Row_Data, 794
- image_key
 - ewl_stock.c, 1208
- image_object
 - Ewl_Button, 589
- image_set
 - Ewl_Stock_Funcs, 752
- img
 - Ewl_Menu_Info, 698
- in_select
 - Ewl_Text, 759
- in_select_mode
 - Ewl_Entry, 635
- increment
 - Ewl_Scrollbar, 731
 - Ewl_Spinner, 747
- index
 - Ewl_Callback_Chain, 594
- inheritance
 - Ewl_Widget, 786
- inmenu
 - Ewl_Menu_Item, 699
- inner_box
 - Ewl_Menubar, 701
- INSET_BOTTOM
 - Ewl_Object, 354
- INSET_HORIZONTAL
 - Ewl_Object, 354
- INSET_LEFT
 - Ewl_Object, 354
- INSET_RIGHT
 - Ewl_Object, 354
- INSET_TOP
 - Ewl_Object, 354
- INSET_VERTICAL
 - Ewl_Object, 354
- insets
 - Ewl_Object, 710
- instance
 - Ewl_Config, 614
- invert
 - Ewl_Range, 727
 - Ewl_Scrollbar, 731
- is
 - Ewl_Model, 703

- item
 - Ewl_Menu, 696
- iterator
 - Ewl_Container, 619
- key
 - Ewl_Image, 687
 - Ewl_Pair, 713
- KEY_COMPARE
 - lib/ewl_text.c, 1223
- keyname
 - Ewl_Event_Key, 644
- l
 - Ewl_Object, 710
- label
 - Ewl_Border, 584
 - Ewl_Icon, 684
 - Ewl_Progressbar, 723
 - ewl_stock.c, 1208
- label_object
 - Ewl_Button, 589
- label_position
 - Ewl_Border, 584
 - Ewl_Checkbutton, 598
- label_set
 - Ewl_Stock_Funcs, 752
- label_text
 - Ewl_Icon, 684
- last
 - Ewl_Embed, 631
 - Ewl_Filelist, 659
- last_grab
 - Ewl_Paned, 715
- last_pos
 - Ewl_Paned, 715
- last_size
 - Ewl_Paned, 715
- last_value
 - Ewl_Spinner, 747
- layer
 - Ewl_Widget, 786
- layout
 - Ewl_Freebox, 673
- left
 - Ewl_Statusbar, 748
- len
 - Ewl_Attach_List, 581
 - Ewl_Callback_Chain, 594
 - Ewl_Event_Dnd_Data_Received, 640
 - Ewl_Pair_List, 714
- length
 - Ewl_Text, 759
- level
 - Ewl_Config_Cache, 615
- lib/Ewl.h, 817
- lib/ewl_attach.c, 821
- lib/ewl_attach.h, 823
- lib/ewl_base.h, 825
- lib/ewl_border.c, 828
- lib/ewl_border.h, 830
- lib/ewl_box.c, 832
 - ewl_box_cb_child_homogeneous_show, 833
 - ewl_box_cb_child_resize, 833
 - ewl_box_cb_configure, 833
 - ewl_box_cb_configure_homogeneous, 833
- lib/ewl_box.h, 835
- lib/ewl_button.c, 837
 - ewl_button_cb_key_down, 838
- lib/ewl_button.h, 840
- lib/ewl_calendar.c, 842
- lib/ewl_calendar.h, 845
- lib/ewl_callback.c, 847
- lib/ewl_callback.h, 849
- lib/ewl_cell.c, 851
- lib/ewl_cell.h, 852
- lib/ewl_check.c, 853
- lib/ewl_check.h, 854
- lib/ewl_checkbutton.c, 856
- lib/ewl_checkbutton.h, 858
- lib/ewl_colordialog.c, 860
 - ewl_colordialog_cb_button_click, 861
 - ewl_colordialog_cb_delete_window, 861
- lib/ewl_colordialog.h, 863
- lib/ewl_colorpicker.c, 865
 - ewl_colorpicker_cb_dnd_data, 868
 - ewl_colorpicker_cb_previous_clicked, 868
 - ewl_colorpicker_cb_radio_change, 868
 - ewl_colorpicker_cb_spinner_change, 868
 - ewl_colorpicker_cb_square_change, 868
 - ewl_colorpicker_cb_vertical_change, 868
 - EWL_COLORPICKER - RADIOBUTTON, 867
 - Ewl_Colorpicker_Radiobutton, 867
 - EWL_COLORPICKER - RADIOBUTTON_TYPE, 867
 - EWL_COLORPICKER_SPINNER, 867
 - Ewl_Colorpicker_Spinner, 867
 - EWL_COLORPICKER_SPINNER - TYPE, 867
- lib/ewl_colorpicker.h, 870
- lib/ewl_combo.c, 872
 - ewl_combo_cb_decrement_clicked, 873
 - ewl_combo_cb_popup_hide, 873
 - ewl_combo_cell_cb_clicked, 873

- ewl_combo_submenu_new, 873
- lib/ewl_combo.h, 875
- lib/ewl_config.c, 877
- lib/ewl_config.h, 879
- lib/ewl_container.c, 881
 - ewl_container_cb_configure, 884
 - ewl_container_cb_disable, 884
 - ewl_container_cb_enable, 884
 - ewl_container_cb_obscure, 884
 - ewl_container_cb_realize, 884
 - ewl_container_cb_reparent, 884
 - ewl_container_cb_reveal, 884
 - ewl_container_cb_unrealize, 884
- lib/ewl_container.h, 886
- lib/ewl_context_menu.c, 890
- lib/ewl_context_menu.h, 892
- lib/ewl_cursor.c, 894
- lib/ewl_cursor.h, 895
- lib/ewl_datepicker.c, 896
 - ewl_datepicker_cb_destroy, 897
 - ewl_datepicker_cb_value_changed, 897
 - ewl_datepicker_cb_window_mouse_down, 897
- lib/ewl_datepicker.h, 899
- lib/ewl_debug.h, 900
- lib/ewl_dialog.c, 906
- lib/ewl_dialog.h, 909
- lib/ewl_dnd.c, 911
- lib/ewl_dnd.h, 913
- lib/ewl_embed.c, 915
- lib/ewl_embed.h, 920
- lib/ewl_engines.c, 924
- lib/ewl_engines.h, 941
- lib/ewl_entry.c, 964
 - ewl_entry_cb_configure, 966
 - ewl_entry_cb_disable, 966
 - ewl_entry_cb_dnd_data, 966
 - ewl_entry_cb_dnd_position, 966
 - ewl_entry_cb_enable, 966
 - ewl_entry_cb_focus_in, 966
 - ewl_entry_cb_focus_out, 966
 - ewl_entry_cb_key_down, 966
 - ewl_entry_cb_mouse_down, 966
 - ewl_entry_cb_mouse_move, 966
 - ewl_entry_cb_mouse_up, 966
- lib/ewl_entry.h, 968
- lib/ewl_enums.h, 971
- lib/ewl_events.c, 977
- lib/ewl_events.h, 978
- lib/ewl_expansion.c, 980
- lib/ewl_expansion.h, 982
- lib/ewl_filedialog.c, 983
 - ewl_filedialog_cb_delete_window, 985
- lib/ewl_filedialog.h, 987
- lib/ewl_filelist.c, 989
- lib/ewl_filelist.h, 992
- lib/ewl_filelist_column.c, 995
- lib/ewl_filelist_column.h, 997
- lib/ewl_filelist_icon.c, 999
- lib/ewl_filelist_icon.h, 1000
- lib/ewl_filelist_list.c, 1002
- lib/ewl_filelist_list.h, 1004
- lib/ewl_filelist_tree.c, 1006
- lib/ewl_filelist_tree.h, 1008
- lib/ewl_filepicker.c, 1010
 - Ewl_Filepicker_Filter, 1012
- lib/ewl_filepicker.h, 1014
- lib/ewl_float.c, 1016
 - ewl_floater_cb_follow_configure, 1017
 - ewl_floater_cb_follow_destroy, 1017
 - ewl_floater_relative_set, 1017
- lib/ewl_float.h, 1019
- lib/ewl_freebox.c, 1021
 - ewl_freebox_cb_configure, 1022
- lib/ewl_freebox.h, 1024
- lib/ewl_grid.c, 1026
 - ewl_grid_cb_configure, 1028
 - ewl_grid_cb_destroy, 1028
- lib/ewl_grid.h, 1030
- lib/ewl_highlight.c, 1033
- lib/ewl_highlight.h, 1034
- lib/ewl_histogram.c, 1036
 - A_CALC, 1037
 - B_CALC, 1037
 - ewl_histogram_cb_configure, 1037
 - G_CALC, 1037
 - R_CALC, 1037
 - Y_CALC, 1037
- lib/ewl_histogram.h, 1039
- lib/ewl_icon.c, 1041
 - ewl_icon_cb_destroy, 1043
 - EWL_ICON_COMPRESS_SIZE, 1043
- lib/ewl_icon.h, 1045
- lib/ewl_icon_theme.c, 1048
- lib/ewl_icon_theme.h, 1050
- lib/ewl_image.c, 1057
 - ewl_image_cb_configure, 1059
 - ewl_image_cb_destroy, 1059
 - ewl_image_cb_mouse_down, 1059
 - ewl_image_cb_mouse_move, 1059
 - ewl_image_cb_mouse_up, 1059
 - ewl_image_cb_obscure, 1059
 - ewl_image_cb_reparent, 1059
 - ewl_image_cb_reveal, 1059
- lib/ewl_image.h, 1061
- lib/ewl_io_manager.c, 1064
 - ewl_io_manager_extension_icon_name_get, 1065

- ewl_io_manager_init, 1065
- ewl_io_manager_mime_type_icon_name_get, 1065
- ewl_io_manager_shutdown, 1065
- ewl_io_manager_string_read, 1065
- ewl_io_manager_string_write, 1066
- ewl_io_manager_uri_mime_type_get, 1066
- ewl_io_manager_uri_read, 1066
- ewl_io_manager_uri_write, 1066
- lib/ewl_io_manager.h, 1069
- lib/ewl_label.c, 1073
- lib/ewl_label.h, 1074
- lib/ewl_list.c, 1076
 - ewl_list_cb_configure, 1076
 - ewl_list_cb_item_clicked, 1076
- lib/ewl_list.h, 1078
- lib/ewl_macros.h, 1079
- lib/ewl_media.c, 1082
 - ewl_media_cb_configure, 1083
 - ewl_media_cb_realize, 1083
 - ewl_media_cb_reveal, 1083
 - ewl_media_cb_unrealize, 1083
- lib/ewl_media.h, 1085
- lib/ewl_menu.c, 1087
 - ewl_menu_cb_configure, 1088
 - ewl_menu_cb_destroy, 1088
 - ewl_menu_cb_expand, 1088
 - ewl_menu_cb_mouse_move, 1088
 - ewl_menu_cb_popup_destroy, 1088
- lib/ewl_menu.h, 1090
- lib/ewl_menu_item.c, 1092
- lib/ewl_menu_item.h, 1093
- lib/ewl_menubar.c, 1095
- lib/ewl_menubar.h, 1097
- lib/ewl_misc.c, 1099
- lib/ewl_misc.h, 1102
- lib/ewl_model.c, 1103
- lib/ewl_model.h, 1106
- lib/ewl_mvc.c, 1109
 - ewl_mvc_cb_destroy, 1111
 - ewl_mvc_selected_is, 1111
 - ewl_mvc_selected_rm, 1111
- lib/ewl_mvc.h, 1113
- lib/ewl_notebook.c, 1117
 - ewl_notebook_cb_tab_clicked, 1118
- lib/ewl_notebook.h, 1120
- lib/ewl_object.c, 1122
- lib/ewl_object.h, 1127
- lib/ewl_overlay.c, 1133
- lib/ewl_overlay.h, 1134
- lib/ewl_paned.c, 1135
 - ewl_paned_cb_configure, 1136
- lib/ewl_paned.h, 1138
- lib/ewl_password.c, 1140
 - ewl_password_cb_destroy, 1141
 - ewl_password_cb_key_down, 1141
- lib/ewl_password.h, 1143
- lib/ewl_popup.c, 1145
- lib/ewl_popup.h, 1147
- lib/ewl_private.h, 1149
- lib/ewl_progressbar.c, 1154
 - ewl_progressbar_cb_configure, 1155
 - ewl_progressbar_cb_value_changed, 1155
- lib/ewl_progressbar.h, 1157
- lib/ewl_radiobutton.c, 1159
- lib/ewl_radiobutton.h, 1160
- lib/ewl_range.c, 1162
- lib/ewl_range.h, 1164
- lib/ewl_row.c, 1166
- lib/ewl_row.h, 1168
- lib/ewl_scrollbar.c, 1170
 - ewl_scrollbar_cb_scroll_start, 1171
 - ewl_scrollbar_cb_scroll_stop, 1171
- lib/ewl_scrollbar.h, 1173
- lib/ewl_scrollpane.c, 1175
 - ewl_scrollpane_cb_configure, 1176
 - ewl_scrollpane_cb_focus_jump, 1176
 - ewl_scrollpane_cb_hscroll, 1176
 - ewl_scrollpane_cb_vscroll, 1176
 - ewl_scrollpane_cb_wheel_scroll, 1176
- lib/ewl_scrollpane.h, 1178
- lib/ewl_seeker.c, 1180
 - ewl_seeker_cb_button_mouse_down, 1181
 - ewl_seeker_cb_button_mouse_up, 1181
 - ewl_seeker_cb_configure, 1181
 - ewl_seeker_cb_key_down, 1181
 - ewl_seeker_cb_mouse_down, 1181
 - ewl_seeker_cb_mouse_move, 1181
 - ewl_seeker_cb_mouse_up, 1181
- lib/ewl_seeker.h, 1183
- lib/ewl_separator.c, 1185
- lib/ewl_separator.h, 1186
- lib/ewl_shadow.c, 1188
- lib/ewl_shadow.h, 1190
- lib/ewl_spacer.c, 1191
- lib/ewl_spacer.h, 1192
- lib/ewl_spectrum.c, 1193
- lib/ewl_spectrum.h, 1195
- lib/ewl_spinner.c, 1197
 - ewl_spinner_cb_configure, 1198
 - ewl_spinner_cb_destroy, 1198
 - ewl_spinner_cb_focus_out, 1198
 - ewl_spinner_cb_key_down, 1198
 - ewl_spinner_cb_realize, 1198
 - ewl_spinner_cb_value_changed, 1198

- ewl_spinner_cb_value_stop, 1198
- ewl_spinner_cb_wheel, 1198
- lib/ewl_spinner.h, 1200
- lib/ewl_statusbar.c, 1202
 - ewl_statusbar_cb_destroy, 1203
- lib/ewl_statusbar.h, 1205
- lib/ewl_stock.c, 1207
- lib/ewl_stock.h, 1209
- lib/ewl_table.c, 1211
 - ewl_table_cb_child_select, 1212
 - ewl_table_cb_child_show, 1212
 - ewl_table_cb_configure, 1212
- lib/ewl_table.h, 1214
- lib/ewl_text.c, 1216
 - COLOR_HASH, 1223
 - ewl_text_cb_configure, 1224
 - ewl_text_cb_destroy, 1224
 - ewl_text_cb_hide, 1224
 - ewl_text_cb_mouse_down, 1224
 - ewl_text_cb_mouse_move, 1224
 - ewl_text_cb_mouse_up, 1224
 - ewl_text_cb_obscure, 1224
 - ewl_text_cb_reveal, 1224
 - ewl_text_cb_show, 1224
 - EWL_TEXT_CHAR_BYTE_LEN, 1223
 - EWL_TEXT_EXTEND_VAL, 1223
 - Ewl_Text_Fmt, 1223
 - ewl_text_trigger_cb_destroy, 1224
 - ewl_text_trigger_cb_mouse_down, 1224
 - ewl_text_trigger_cb_mouse_in, 1224
 - ewl_text_trigger_cb_mouse_out, 1224
 - ewl_text_trigger_cb_mouse_up, 1224
 - KEY_COMPARE, 1223
- lib/ewl_text.h, 1226
- lib/ewl_theme.c, 1234
 - ewl_embed_list, 1235
- lib/ewl_theme.h, 1237
- lib/ewl_toolbar.c, 1238
- lib/ewl_toolbar.h, 1240
- lib/ewl_tree.c, 1242
 - ewl_tree_cb_configure, 1245
 - ewl_tree_cb_destroy, 1245
 - ewl_tree_cb_header_change, 1245
 - ewl_tree_cb_header_configure, 1245
 - ewl_tree_cb_hscroll, 1245
 - ewl_tree_cb_node_child_add, 1245
 - ewl_tree_cb_node_child_show, 1245
 - ewl_tree_cb_node_configure, 1245
 - ewl_tree_cb_node_destroy, 1245
 - ewl_tree_cb_node_toggle, 1245
 - ewl_tree_cb_row_hide, 1245
 - ewl_tree_cb_row_select, 1245
 - ewl_tree_row_remove, 1245
- lib/ewl_tree.h, 1247
- lib/ewl_tree2.c, 1250
 - ewl_tree2_cb_column_sort, 1252
 - ewl_tree2_cb_configure, 1252
 - ewl_tree2_cb_destroy, 1252
 - ewl_tree2_cb_node_child_add, 1252
 - ewl_tree2_cb_node_child_show, 1252
 - ewl_tree2_cb_node_configure, 1252
 - ewl_tree2_cb_node_toggle, 1252
- lib/ewl_tree2.h, 1255
- lib/ewl_tree2_view.c, 1258
- lib/ewl_tree2_view.h, 1259
- lib/ewl_tree2_view_plain.c, 1261
- lib/ewl_tree2_view_plain.h, 1262
- lib/ewl_tree2_view_scrolled.c, 1263
- lib/ewl_tree2_view_scrolled.h, 1264
- lib/ewl_view.c, 1266
- lib/ewl_view.h, 1268
- lib/ewl_widget.c, 1270
 - ewl_widget_cb_configure, 1275
 - ewl_widget_cb_disable, 1275
 - ewl_widget_cb_drag_down, 1275
 - ewl_widget_cb_drag_move, 1275
 - ewl_widget_cb_drag_up, 1275
 - ewl_widget_cb_enable, 1275
 - ewl_widget_cb_focus_in, 1275
 - ewl_widget_cb_focus_out, 1275
 - ewl_widget_cb_hide, 1275
 - ewl_widget_cb_mouse_down, 1275
 - ewl_widget_cb_mouse_in, 1275
 - ewl_widget_cb_mouse_move, 1275
 - ewl_widget_cb_mouse_out, 1275
 - ewl_widget_cb_mouse_up, 1275
 - ewl_widget_cb_obscure, 1275
 - ewl_widget_cb_realize, 1275
 - ewl_widget_cb_reparent, 1275
 - ewl_widget_cb_reveal, 1275
 - ewl_widget_cb_show, 1275
 - ewl_widget_cb_unrealize, 1275
 - ewl_widget_new, 1275
- lib/ewl_widget.h, 1278
- lib/ewl_window.c, 1284
- lib/ewl_window.h, 1288
- list
 - Ewl_Attach_List, 581
 - Ewl_Callback_Chain, 594
 - Ewl_Filelist_Column, 661
 - Ewl_Filelist_Column_Data, 663
 - Ewl_Filelist_Icon, 664
 - Ewl_Filelist_List, 665
 - Ewl_Filelist_Tree, 666
 - Ewl_Filelist_Tree_Data, 667
 - Ewl_Pair_List, 714
 - List_Test_Data, 791
 - bin/tests/list/ewl_list.c, 1077

- count, 791
- rows, 791
- List_Test_Row_Data, 792
 - bin/tests/list/ewl_list.c, 1077
 - image, 792
 - text, 792
- main
 - ewl_embed_test.c, 796
 - ewl_simple_test.c, 797
 - main.c, 805
- main.c
 - main, 805
 - MAIN_HEIGHT, 805
 - MAIN_WIDTH, 805
- MAIN_HEIGHT
 - main.c, 805
- MAIN_WIDTH
 - main.c, 805
- map
 - Ewl_Grid, 676
- mask
 - Ewl_Callback_Chain, 594
- MAX
 - Ewl_Macros, 293
- max_val
 - Ewl_Range, 727
- maximum
 - Ewl_Object, 710
- MAXIMUM_H
 - Ewl_Object, 354
- MAXIMUM_W
 - Ewl_Object, 354
- maxv
 - Ewl_Histogram, 682
- media
 - Ewl_Media, 694
- menu
 - Ewl_Filedialog, 657
 - Ewl_Icon, 684
 - Ewl_Menubar_Info, 702
- menubar
 - Ewl_Toolbar, 768
- menubar_parent
 - Ewl_Menu, 696
- MIN
 - Ewl_Macros, 293
- min_val
 - Ewl_Range, 727
- minimum
 - Ewl_Object, 710
- MINIMUM_H
 - Ewl_Object, 354
- minimum_size
 - Ewl_Paned_Layout, 718
- MINIMUM_W
 - Ewl_Object, 354
- mode
 - Ewl_Colorpicker, 604
 - Ewl_Colorpicker_Radiobutton, 607
 - Ewl_Colorpicker_Spinner, 609
 - Ewl_Spectrum, 744
 - Ewl_Tree, 770
- model
 - Ewl_Combo_Cell, 612
 - Ewl_Model, 704
 - Ewl_MVC, 706
 - Ewl_Selection, 736
- modifiers
 - Ewl_Event_Key, 644
 - Ewl_Event_Mouse, 647
- module
 - Ewl_Media, 694
- month_label
 - Ewl_Calendar, 592
- mouse
 - Ewl_Popup, 722
- mouse_in
 - Ewl_Embed, 631
- multiline
 - Ewl_Entry, 636
- multiselect
 - Ewl_Filelist, 659
- multiselect_change
 - Ewl_Filelist, 659
- mute
 - Ewl_Media, 694
- mvc
 - Ewl_Combo, 611
 - Ewl_List, 693
 - Ewl_Tree2, 772
 - Ewl_Tree2_Node, 775
- mvc_data
 - Ewl_Combo_Cell, 613
- name
 - Control, 578
 - Ewl_Engine, 633
 - Ewl_Filepicker_Filter, 670
 - Ewl_Menu_Info, 698
 - Ewl_Menubar_Info, 702
 - Ewl_Test, 755
 - Ewl_Unit_Test, 783
 - Ewl_Window, 789
 - Freebox_Test, 790
- ncols
 - Ewl_Tree, 770
- NEST

- bin/tests/tree/ewl_tree.c, 1246
- NEW
 - Ewl_Macros, 293
- new_panes
 - Ewl_Paned, 716
- nodes
 - Ewl_Text, 759
- num_types
 - Ewl_Dnd_Types, 628
- obj_cache
 - Ewl_Embed, 631
- object
 - Ewl_Widget, 786
- obscure
 - Ewl_Password, 719
- OBSCURED
 - Ewl_Object, 354
- offset
 - Ewl_Popup, 722
 - Ewl_Text, 759
- oh
 - Ewl_Image, 687
- old_flags
 - Ewl_Window, 789
- open_menu
 - Ewl_Context_Menu, 621
- orientation
 - Ewl_Box, 586
 - Ewl_Freebox, 673
 - Ewl_Grid, 676
 - Ewl_Paned, 716
 - Ewl_Seeker, 735
 - Ewl_Separator, 740
- orig
 - Ewl_Image_Thumbnail, 689
- outer_box
 - Ewl_Menubar, 701
 - Ewl_Statusbar, 748
- outline
 - Ewl_Text_Context, 762
- overlay
 - Ewl_Embed, 631
 - Ewl_Scrollpane, 733
 - Ewl_Spectrum, 744
- ow
 - Ewl_Image, 687
- pad
 - Ewl_Object, 711
- PADDING_BOTTOM
 - Ewl_Object, 355
- PADDING_HORIZONTAL
 - Ewl_Object, 355
- PADDING_LEFT
 - Ewl_Object, 355
- PADDING_RIGHT
 - Ewl_Object, 355
- PADDING_TOP
 - Ewl_Object, 355
- PADDING_VERTICAL
 - Ewl_Object, 355
- pages
 - Ewl_Notebook, 708
- parent
 - Ewl_Entry_Cursor, 637
 - Ewl_Tree2_View, 776
 - Ewl_Widget, 786
- path
 - Ewl_Filepicker, 669
 - Ewl_Image, 687
- path_combo
 - Ewl_Filepicker, 669
- PATH_MAX
 - bin/tests/media/ewl_media.c, 1084
 - ewl_private.h, 1153
- picker
 - Ewl_Colordialog, 601
 - Ewl_Colorpicker, 605
- placed
 - Ewl_Paned_Grabber, 717
- pointer
 - Ewl_Engine_Info, 634
- popup
 - Ewl_Combo, 611
 - Ewl_Context_Menu, 621
 - Ewl_Menu, 697
- position
 - Ewl_Dialog, 627
 - Ewl_Media, 695
- position_request
 - Ewl_Paned_Layout, 718
- pref_fill_set
 - Ewl_Box_Orientation, 588
- preferred
 - Ewl_Object, 711
- PREFERRED_H
 - Ewl_Object, 355
- preferred_size
 - Ewl_Grid_Info, 678
 - Ewl_Paned_Layout, 718
- PREFERRED_W
 - Ewl_Object, 355
- preview
 - Ewl_Colorpicker, 605
 - Ewl_Filelist_Column, 662
- previous
 - Ewl_Colorpicker, 605

- print_keys
 - Ewl_Config_Cache, 615
- print_signals
 - Ewl_Config_Cache, 615
- priv_data
 - Ewl_Attach, 579
- proportional
 - Ewl_Image, 687
- r
 - Ewl_Color_Set, 600
 - Ewl_Colorpicker, 605
 - Ewl_Object, 711
- R_CALC
 - lib/ewl_histogram.c, 1037
- range
 - Ewl_Progressbar, 723
 - Ewl_Seeker, 735
 - Ewl_Spinner, 747
- real_text
 - Ewl_Password, 719
- REALIZED
 - Ewl_Object, 355
- REALLOC
 - Ewl_Macros, 293
- RECURSIVE
 - Ewl_Object, 355
- redirect
 - Ewl_Container, 619
- ref_count
 - Ewl_Text_Context, 762
- refcount
 - Ewl_Cursor, 622
- references
 - Ewl_Callback, 593
- rel_size
 - Ewl_Grid_Info, 678
- resize_type
 - Ewl_Grid_Info, 678
- response
 - Ewl_Event_Action_Response, 639
- rgb
 - Ewl_Colorpicker, 605
 - Ewl_Spectrum, 744
- right
 - Ewl_Statusbar, 749
- row
 - Ewl_Selection_Idx, 737
 - Ewl_Selection_Range, 738
 - Ewl_Tree2_Node, 775
 - Ewl_Tree_Node, 781
- row_num
 - Ewl_Tree2_Node, 775
- row_select
 - Ewl_Table, 754
- row_size
 - Ewl_Grid, 676
- rowcache
 - Ewl_Tree2, 772
- ROWS
 - bin/tests/tree/ewl_tree.c, 1246
- rows
 - Ewl_Grid, 676
 - Ewl_Tree2, 772
 - List_Test_Data, 791
 - Tree2_Test_Data, 793
- s
 - Ewl_Colorpicker, 605
 - Ewl_Spectrum, 744
- scroll
 - Ewl_Tree2_View_Scrolled, 779
- scroll_flags
 - Ewl_Filelist, 660
- scroll_header
 - Ewl_Tree2_View_Scrolled, 779
- scroll_visible
 - Ewl_Tree2_View_Scrolled, 779
- scrollable
 - Ewl_Combo, 611
- scrollarea
 - Ewl_Tree, 770
- seeker
 - Ewl_Scrollbar, 731
- segv
 - Ewl_Config_Cache, 616
- sel
 - Ewl_Selection_Idx, 737
 - Ewl_Selection_Range, 738
- select
 - Ewl_Filelist, 660
- selectable
 - Ewl_Text, 759
- selected
 - Ewl_Filelist, 660
 - Ewl_MVC, 706
 - Ewl_Table, 754
 - Ewl_Tree, 770
- selected_change
 - Ewl_MVC, 706
- selected_file_add
 - Ewl_Filelist, 660
- selected_unselect
 - Ewl_Filelist, 660
- selection
 - Ewl_Text, 759
- selection_mode
 - Ewl_MVC, 706

- separator
 - Ewl_Dialog, 627
 - Ewl_Paned_Grabber, 717
- set
 - Ewl_Image, 687
- sh
 - Ewl_Image, 687
- shadow
 - Ewl_Text_Context, 762
- shift_handle
 - Ewl_Filelist, 660
- show_dot_change
 - Ewl_Filelist, 660
- show_dot_files
 - Ewl_Filelist, 660
- show_favorites
 - Ewl_Filepicker, 669
- shutdown
 - Ewl_Engine_Info, 634
- size
 - Ewl_Attach_Dnd, 580
 - Ewl_Grid_Info, 678
 - Ewl_Text_Context, 762
- smart
 - Ewl_Embed, 631
- smart_object
 - Ewl_Widget, 786
- sort
 - Ewl_Model, 704
 - Ewl_Tree2, 772
- sortable
 - Ewl_Model, 704
- sorted
 - Ewl_Freebox, 674
- source
 - Ewl_Histogram, 682
- space
 - Ewl_Grid, 676
- spacing
 - Ewl_Box, 586
- spinner
 - Ewl_Colorpicker_Spinner, 609
- spinners
 - Ewl_Colorpicker, 605
- square
 - Ewl_Colorpicker, 605
- stable_position_request
 - Ewl_Paned_Layout, 718
- stable_request
 - Ewl_Paned_Layout, 718
- stack
 - Ewl_Statusbar, 749
- start
 - Ewl_Selection_Range, 739
- start_c
 - Ewl_Table, 754
- start_col
 - Ewl_Grid_Child, 677
- start_r
 - Ewl_Table, 754
- start_row
 - Ewl_Grid_Child, 677
- start_time
 - Ewl_Scrollbar, 731
 - Ewl_Seeker, 735
 - Ewl_Spinner, 747
- status
 - Ewl_Statusbar, 749
 - Ewl_Unit_Test, 783
- step
 - Ewl_Range, 728
- stock
 - Ewl_Button, 590
 - Ewl_Icon, 684
- stock_funcs
 - Ewl_Stock, 750
- stock_type
 - Ewl_Stock, 750
- strikethrough
 - Ewl_Text_Context, 762
- string_read
 - Ewl_IO_Manager_Plugin, 690
- string_write
 - Ewl_IO_Manager_Plugin, 690
- style_colors
 - Ewl_Text_Context, 762
- styles
 - Ewl_Text_Context, 763
- subdata
 - Tree2_Test_Row_Data, 794
- sw
 - Ewl_Image, 687
- system
 - Ewl_Config, 614
- t
 - Ewl_Object, 711
- tab_order
 - Ewl_Embed, 631
- tabbar
 - Ewl_Notebook, 708
- tabbar_position
 - Ewl_Notebook, 708
- test_info
 - bin/tests/border/ewl_border.c, 829
 - bin/tests/box/ewl_box.c, 834
 - bin/tests/button/ewl_button.c, 839
 - bin/tests/calendar/ewl_calendar.c, 844

- bin/tests/colordialog/ewl_colordialog.c, 862
- bin/tests/colorpicker/ewl_colorpicker.c, 869
- bin/tests/combo/ewl_combo.c, 874
- bin/tests/container/ewl_container.c, 885
- bin/tests/datepicker/ewl_datepicker.c, 898
- bin/tests/dialog/ewl_dialog.c, 908
- bin/tests/entry/ewl_entry.c, 967
- bin/tests/filedialog/ewl_filedialog.c, 986
- bin/tests/filepicker/ewl_filepicker.c, 1013
- bin/tests/floater/ewl_floater.c, 1018
- bin/tests/freebox/ewl_freebox.c, 1023
- bin/tests/grid/ewl_grid.c, 1029
- bin/tests/histogram/ewl_histogram.c, 1038
- bin/tests/icon/ewl_icon.c, 1044
- bin/tests/icon_theme/ewl_icon_theme.c, 1049
- bin/tests/image/ewl_image.c, 1060
- bin/tests/io_manager/ewl_io_manager.c, 1068
- bin/tests/list/ewl_list.c, 1077
- bin/tests/media/ewl_media.c, 1084
- bin/tests/menu/ewl_menu.c, 1089
- bin/tests/menubar/ewl_menubar.c, 1096
- bin/tests/mvc/ewl_mvc.c, 1112
- bin/tests/notebook/ewl_notebook.c, 1119
- bin/tests/paned/ewl_paned.c, 1137
- bin/tests/password/ewl_password.c, 1142
- bin/tests/progressbar/ewl_progressbar.c, 1156
- bin/tests/scrollbar/ewl_scrollbar.c, 1172
- bin/tests/scrollpane/ewl_scrollpane.c, 1177
- bin/tests/seeker/ewl_seeker.c, 1182
- bin/tests/shadow/ewl_shadow.c, 1189
- bin/tests/spinner/ewl_spinner.c, 1199
- bin/tests/statusbar/ewl_statusbar.c, 1204
- bin/tests/table/ewl_table.c, 1213
- bin/tests/text/ewl_text.c, 1225
- bin/tests/theme/ewl_theme.c, 1236
- bin/tests/toolbar/ewl_toolbar.c, 1239
- bin/tests/tree/ewl_tree.c, 1246
- bin/tests/tree2/ewl_tree2.c, 1254
- bin/tests/widget/ewl_widget.c, 1277
- ewl_dnd_snoop.c, 807
- ewl_engine.c, 808
- ewl_fullscreen.c, 809
- ewl_image_thumbnail.c, 810
- ewl_layer.c, 811
- ewl_modal.c, 812
- ewl_pointer.c, 813
- ewl_puzzle.c, 814
- ewl_text_editor.c, 815
- ewl_tooltip.c, 816
- text
 - Ewl_Datepicker, 624
 - ewl_embed_test.c, 796
 - Ewl_Entry, 636
 - Ewl_Text, 759
 - List_Test_Row_Data, 792
 - Tree2_Test_Row_Data, 794
- text_parent
 - Ewl_Text_Trigger, 766
- textblock
 - Ewl_Text, 759
- theme
 - Ewl_Engine_Info, 634
 - Ewl_Widget, 786
- theme_group
 - Ewl_Widget, 786
- theme_object
 - Ewl_Widget, 786
- theme_path
 - Ewl_Widget, 786
- theme_state
 - Ewl_Widget, 786
- theme_text
 - Ewl_Widget, 787
- thumb
 - Ewl_Image_Thumbnail, 689
- thumbnailing
 - Ewl_Icon, 684
- tile
 - Ewl_Image, 687
- timer
 - Ewl_Attach_Tooltip, 583
 - Ewl_Scrollbar, 731
 - Ewl_Seeker, 735
 - Ewl_Spinner, 747
- tip
 - Ewl_Test, 755
- title
 - Ewl_Window, 789
- to
 - Ewl_Attach_Tooltip, 583
- tooltip
 - ewl_stock.c, 1208
- tooltip_set
 - Ewl_Stock_Funcs, 752
- toplayered
 - Ewl_Widget, 787
- total_size
 - Ewl_Text, 759
- transient
 - Ewl_Window, 789

- tree
 - Ewl_Filelist_List, 665
 - Ewl_Filelist_Tree, 666
 - Ewl_Tree2_Node, 775
 - Ewl_Tree_Node, 782
- TREE2_DATA_ELEMENTS
 - bin/tests/tree2/ewl_tree2.c, 1254
- Tree2_Test_Data, 793
 - bin/tests/tree2/ewl_tree2.c, 1254
 - count, 793
 - rows, 793
- Tree2_Test_Row_Data, 794
 - bin/tests/tree2/ewl_tree2.c, 1254
 - expandable, 794
 - image, 794
 - subdata, 794
 - text, 794
- triggers
 - Ewl_Text, 759
- tx
 - Ewl_Text, 759
 - Ewl_Text_Fmt, 764
- type
 - Ewl_Attach, 579
 - Ewl_Event_Dnd_Data_Received, 640
 - Ewl_Event_Dnd_Data_Request, 641
 - Ewl_Icon, 685
 - Ewl_Image, 688
 - Ewl_Popup, 722
 - Ewl_Selection, 736
 - Ewl_Spectrum, 744
 - Ewl_Test, 755
 - Ewl_Text_Trigger, 766
 - Ewl_Tree2, 773
 - Freebox_Test, 790
- type_combo
 - Ewl_Filepicker, 669
- types
 - Ewl_Dnd_Types, 628
- underline
 - Ewl_Text_Context, 763
- unit_tests
 - Ewl_Test, 756
- updating
 - Ewl_Colorpicker, 605
- updating_grabbers
 - Ewl_Paned, 716
- uri_read
 - Ewl_IO_Manager_Plugin, 690
- uri_write
 - Ewl_IO_Manager_Plugin, 690
- user
 - Ewl_Config, 614
 - Ewl_Grid_Info, 678
- user_data
 - Ewl_Callback, 593
- v
 - Ewl_Colorpicker, 605
 - Ewl_Filelist, 660
 - Ewl_Spectrum, 744
- value
 - Ewl_Pair, 713
 - Ewl_Range, 728
- variable_request
 - Ewl_Paned_Layout, 718
- vbox
 - Ewl_Dialog, 627
 - Ewl_Spinner, 747
- vertical
 - Ewl_Colorpicker, 605
 - Ewl_Spectrum, 745
- vflag
 - Ewl_Scrollpane, 733
- video
 - Ewl_Media, 695
- view
 - Ewl_Filepicker, 669
 - Ewl_MVC, 706
 - Ewl_Tree2_View_Plain, 778
- view_change
 - Ewl_MVC, 706
- VISIBLE
 - Ewl_Object, 355
- volume
 - Ewl_Media, 695
- vscrollbar
 - Ewl_Scrollpane, 733
- w
 - Ewl_Check, 597
 - Ewl_Event_Window_Expose, 654
 - Ewl_Expansion, 655
 - Ewl_Image, 688
 - Ewl_Object, 711
- widget
 - Ewl_Container, 619
 - Ewl_Entry_Cursor, 637
 - Ewl_Image, 688
 - Ewl_Label, 692
 - Ewl_Media, 695
 - Ewl_Separator, 740
 - Ewl_Spacer, 742
 - Ewl_Text_Trigger, 766
 - Ewl_Text_Trigger_Area, 767
- win
 - Ewl_Attach_Tooltip, 583

WIN_NAME

ewl_engine.c, 808

window

Ewl_Cursor, 623

Ewl_Dialog, 627

Ewl_Engine_Info, 634

Ewl_Popup, 722

Ewl_Window, 789

wrap

Ewl_Text_Context, 763

x

Ewl_Attach_Tooltip, 583

Ewl_Embed, 631

Ewl_Event_Dnd_Drop, 642

Ewl_Event_Dnd_Position, 643

Ewl_Event_Mouse, 647

Ewl_Event_Window_Expose, 654

Ewl_Floater, 671

Ewl_Image, 688

Ewl_Object, 711

Ewl_Popup, 722

Ewl_Text, 760

y

Ewl_Attach_Tooltip, 583

Ewl_Embed, 631

Ewl_Event_Dnd_Drop, 642

Ewl_Event_Dnd_Position, 643

Ewl_Event_Mouse, 647

Ewl_Event_Window_Expose, 654

Ewl_Floater, 671

Ewl_Image, 688

Ewl_Object, 711

Ewl_Popup, 722

Ewl_Text, 760

Y_CALC

lib/ewl_histogram.c, 1037

z

Ewl_Event_Mouse_Wheel, 653

ZERO

Ewl_Macros, 293