

# PHPBoost Framework Documentation



# Contents

<a href="#">Package builder Procedural Elements</a>	2
<a href="#">form_builder.class.php</a>	2
<a href="#">Define FIELD INPUT CHECKBOX</a>	3
<a href="#">Define FIELD INPUT FILE</a>	3
<a href="#">Define FIELD INPUT HIDDEN</a>	3
<a href="#">Define FIELD INPUT RADIO</a>	3
<a href="#">Define FIELD INPUT TEXT</a>	3
<a href="#">Define FIELD SELECT</a>	3
<a href="#">Define FIELD TEXTAREA</a>	3
<a href="#">Package builder Classes</a>	4
<a href="#">Class FormBuilder</a>	4
<a href="#">Var \$display_preview</a>	4
<a href="#">Var \$display_reset</a>	4
<a href="#">Var \$field_identifier_preview</a>	4
<a href="#">Var \$form_action</a>	4
<a href="#">Var \$form_class</a>	4
<a href="#">Var \$form_fieldsets</a>	4
<a href="#">Var \$form_name</a>	4
<a href="#">Var \$form_submit</a>	5
<a href="#">Constructor FormBuilder</a>	5
<a href="#">Method add_fieldset</a>	5
<a href="#">Method display</a>	5
<a href="#">Method display_preview_button</a>	5
<a href="#">Method display_reset</a>	5
<a href="#">Method get_form_action</a>	6
<a href="#">Method get_form_class</a>	6
<a href="#">Method get_form_name</a>	6
<a href="#">Method get_form_submit</a>	6
<a href="#">Method set_form_action</a>	6
<a href="#">Method set_form_class</a>	6
<a href="#">Method set_form_name</a>	6
<a href="#">Method set_form_submit</a>	6
<a href="#">Class FormCheckbox</a>	6
<a href="#">Var \$field_options</a>	7
<a href="#">Constructor FormCheckbox</a>	7
<a href="#">Method add_option</a>	7
<a href="#">Method display</a>	7
<a href="#">Class FormCheckboxOption</a>	7
<a href="#">Var \$option_checked</a>	7
<a href="#">Var \$option_title</a>	8
<a href="#">Constructor FormCheckboxOption</a>	8
<a href="#">Method display</a>	8
<a href="#">Class FormField</a>	8

<a href="#">Var \$field_css_class</a>	8
<a href="#">Var \$field_errors</a>	8
<a href="#">Var \$field_id</a>	8
<a href="#">Var \$field_name</a>	9
<a href="#">Var \$field_on_blur</a>	9
<a href="#">Var \$field_required</a>	9
<a href="#">Var \$field_required_alert</a>	9
<a href="#">Var \$field_sub_title</a>	9
<a href="#">Var \$field_title</a>	9
<a href="#">Var \$field_value</a>	9
<a href="#">Constructor FormField</a>	9
<a href="#">Method add_errors</a>	9
<a href="#">Method get_errors</a>	9
<a href="#">Method get_id</a>	10
<a href="#">Method get_required_alert</a>	10
<a href="#">Method throw_error</a>	10
<a href="#">Class FormFieldset</a>	10
<a href="#">Var \$fieldset_display_required</a>	10
<a href="#">Var \$fieldset_errors</a>	10
<a href="#">Var \$fieldset_fields</a>	10
<a href="#">Var \$fieldset_title</a>	10
<a href="#">Constructor FormFieldset</a>	10
<a href="#">Method add_field</a>	11
<a href="#">Method display</a>	11
<a href="#">Method display_preview_button</a>	11
<a href="#">Method get_display_required</a>	11
<a href="#">Method get_fields</a>	11
<a href="#">Method get_title</a>	11
<a href="#">Method set_display_required</a>	11
<a href="#">Method set_title</a>	11
<a href="#">Method throw_error</a>	12
<a href="#">Class FormFileUploader</a>	12
<a href="#">Var \$field_size</a>	12
<a href="#">Constructor FormFileUploader</a>	12
<a href="#">Method display</a>	13
<a href="#">Class FormHiddenField</a>	13
<a href="#">Constructor FormHiddenField</a>	13
<a href="#">Method display</a>	13
<a href="#">Class FormRadioChoice</a>	13
<a href="#">Var \$field_options</a>	13
<a href="#">Constructor FormRadioChoice</a>	14
<a href="#">Method add_option</a>	14
<a href="#">Method display</a>	14
<a href="#">Class FormRadioChoiceOption</a>	14
<a href="#">Var \$option_checked</a>	14
<a href="#">Var \$option_title</a>	15
<a href="#">Constructor FormRadioChoiceOption</a>	15
<a href="#">Method display</a>	15
<a href="#">Class FormSelect</a>	15

<a href="#">Var \$field_multiple</a>	15
<a href="#">Var \$field_options</a>	15
<a href="#">Constructor FormSelect</a>	15
<a href="#">Method add_option</a>	15
<a href="#">Method display</a>	16
<a href="#">Class FormSelectOption</a>	16
<a href="#">Var \$option_selected</a>	16
<a href="#">Var \$option_title</a>	16
<a href="#">Constructor FormSelectOption</a>	16
<a href="#">Method display</a>	16
<a href="#">Class FormTextarea</a>	16
<a href="#">Var \$field_cols</a>	17
<a href="#">Var \$field_editor</a>	17
<a href="#">Var \$field_forbidden_tags</a>	17
<a href="#">Var \$field_rows</a>	17
<a href="#">Constructor FormTextarea</a>	17
<a href="#">Method display</a>	17
<a href="#">Class FormTextEdit</a>	17
<a href="#">Var \$field_maxlength</a>	18
<a href="#">Var \$field_size</a>	18
<a href="#">Constructor FormTextEdit</a>	18
<a href="#">Method display</a>	18

<a href="#">Package content Procedural Elements</a>	20
<a href="#">categories_manager.class.php</a>	20
<a href="#">Define ADD_THIS_CATEGORY_IN_LIST</a>	20
<a href="#">Define AJAX_MODE</a>	20
<a href="#">Define CATEGORY_DOES_NOT_EXIST</a>	20
<a href="#">Define CAT_UNVISIBLE</a>	20
<a href="#">Define CAT_VISIBLE</a>	20
<a href="#">Define DEBUG_MODE</a>	20
<a href="#">Define DISPLAYING_CONFIGURATION_NOT_SET</a>	20
<a href="#">Define DO_NOT_ADD_THIS_CATEGORY_IN_LIST</a>	20
<a href="#">Define DO_NOT_LOAD_CACHE</a>	20
<a href="#">Define ERROR_CAT_IS_AT_BOTTOM</a>	20
<a href="#">Define ERROR_CAT_IS_AT_TOP</a>	20
<a href="#">Define ERROR_UNKNOWN_MOTION</a>	20
<a href="#">Define</a>	
<a href="#">IGNORE_AND_CONTINUE_BROWSING_IF_A_CATEGORY_DOES_NOT_MA</a>	
<a href="#">TCH</a>	20
<a href="#">Define INCORRECT_DISPLAYING_CONFIGURATION</a>	20
<a href="#">Define LOAD_CACHE</a>	20
<a href="#">Define MOVE_CATEGORY_DOWN</a>	20
<a href="#">Define MOVE_CATEGORY_UP</a>	20
<a href="#">Define NEW_CATEGORY_IS_IN_ITS_CHILDRENS</a>	20
<a href="#">Define NEW_PARENT_CATEGORY_DOES_NOT_EXIST</a>	20
<a href="#">Define NEW_STATUS_UNKNOWN</a>	20
<a href="#">Define NORMAL_MODE</a>	20
<a href="#">Define NOT_RECURSIVE_EXPLORATION</a>	20
<a href="#">Define PRODUCTION_MODE</a>	20

<a href="#">Define RECURSIVE_EXPLORATION</a>	20
<a href="#">Define STOP_BROWSING_IF_A_CATEGORY_DOES_NOT_MATCH</a>	20
<a href="#">comments.class.php</a>	21
<a href="#">Define INTEGRATED_IN_ENVIRONMENT</a>	21
<a href="#">Define KERNEL_SCRIPT</a>	21
<a href="#">Define POP_UP_WINDOW</a>	21
<a href="#">note.class.php</a>	22
<a href="#">Define NOTE_DISPLAY_BLOCK</a>	22
<a href="#">Define NOTE_DISPLAY_NOTE</a>	22
<a href="#">Define NOTE_NODISPLAY_NBRNOTES</a>	22
<a href="#">search.class.php</a>	23
<a href="#">Define CACHE_TIME</a>	23
<a href="#">Define CACHE_TIMES_USED</a>	23
<b><a href="#">Package content Classes</a></b>	<b>24</b>
<a href="#">Class CategoriesManager</a>	24
<a href="#">Var \$cache_file_name</a>	25
<a href="#">Var \$cache_var</a>	25
<a href="#">Var \$display_config</a>	26
<a href="#">Var \$errors</a>	26
<a href="#">Var \$table</a>	26
<a href="#">Constructor CategoriesManager</a>	26
<a href="#">Method add</a>	26
<a href="#">Method build_administration_interface</a>	27
<a href="#">Method build_children_id_list</a>	27
<a href="#">Method build_parents_id_list</a>	28
<a href="#">Method build_select_form</a>	28
<a href="#">Method change_visibility</a>	28
<a href="#">Method check_display_config</a>	29
<a href="#">Method check_error</a>	29
<a href="#">Method compute_heritated_auth</a>	29
<a href="#">Method delete</a>	29
<a href="#">Method get_feeds_list</a>	30
<a href="#">Method move</a>	30
<a href="#">Method move_into_another</a>	30
<a href="#">Method set_display_config</a>	30
<a href="#">Class Comments</a>	31
<a href="#">Var \$idcom</a>	31
<a href="#">Var \$idprov</a>	31
<a href="#">Var \$is_kernel_script</a>	31
<a href="#">Var \$lock_com</a>	31
<a href="#">Var \$module_folder</a>	31
<a href="#">Var \$nbr_com</a>	31
<a href="#">Var \$path</a>	31
<a href="#">Var \$script</a>	31
<a href="#">Var \$sql_table</a>	32
<a href="#">Var \$vars</a>	32
<a href="#">Constructor Comments</a>	32
<a href="#">Method add</a>	32
<a href="#">Method com_display_link</a>	32

<a href="#">Method del</a>	33
<a href="#">Method delete_all</a>	33
<a href="#">Method display</a>	33
<a href="#">Method get_attribute</a>	33
<a href="#">Method is_loaded</a>	33
<a href="#">Method lock</a>	33
<a href="#">Method set_arg</a>	34
<a href="#">Method update</a>	34
<a href="#">Class Note</a>	34
<a href="#">Var \$idprov</a>	34
<a href="#">Var \$module_folder</a>	34
<a href="#">Var \$notation_scale</a>	34
<a href="#">Var \$options</a>	35
<a href="#">Var \$path</a>	35
<a href="#">Var \$script</a>	35
<a href="#">Var \$script_path</a>	35
<a href="#">Var \$sql_table</a>	35
<a href="#">Constructor Note</a>	35
<a href="#">Method add</a>	35
<a href="#">Method display_form</a>	35
<a href="#">Method display_img</a>	36
<a href="#">Method get_attribute</a>	36
<a href="#">Class Search</a>	36
<a href="#">Var \$errors</a>	37
<a href="#">Var \$id_search</a>	37
<a href="#">Var \$id_user</a>	37
<a href="#">Var \$modules</a>	37
<a href="#">Var \$modules_conditions</a>	37
<a href="#">Var \$search</a>	37
<a href="#">Constructor Search</a>	37
<a href="#">Method get_ids</a>	37
<a href="#">Method get_results</a>	37
<a href="#">Method get_results_by_id</a>	38
<a href="#">Method insert_results</a>	38
<a href="#">Method is_in_cache</a>	38
<a href="#">Method is_search_id_in_cache</a>	38
<a href="#">Method modules_in_cache</a>	39
<a href="#">Class BBCodeEditor</a>	39
<a href="#">Constructor BBCodeEditor</a>	39
<a href="#">Method display</a>	39
<a href="#">Class ContentEditor</a>	39
<a href="#">Var \$forbidden_tags</a>	40
<a href="#">Var \$identifier</a>	40
<a href="#">Var \$language_type</a>	40
<a href="#">Var \$template</a>	40
<a href="#">Constructor ContentEditor</a>	40
<a href="#">Method get_forbidden_tags</a>	40
<a href="#">Method get_template</a>	40
<a href="#">Method set_forbidden_tags</a>	40

<a href="#">Method set identifier</a>	40
<a href="#">Method set template</a>	40
<a href="#">Class TinyMCEEditor</a>	41
<a href="#">Var \$array_tags</a>	41
<a href="#">Var \$array_tags2</a>	41
<a href="#">Var \$array_tags3</a>	41
<a href="#">Constructor TinyMCEEditor</a>	41
<a href="#">Method display</a>	41
<a href="#">bbcode_highlighter.class.php</a>	42
<a href="#">Define BBCODE HIGHLIGHTER BLOCK CODE</a>	42
<a href="#">Define BBCODE HIGHLIGHTER INLINE CODE</a>	42
<a href="#">Define BBCODE LIST ITEM COLOR</a>	42
<a href="#">Define BBCODE PARAM COLOR</a>	42
<a href="#">Define BBCODE PARAM NAME COLOR</a>	42
<a href="#">Define BBCODE TAG COLOR</a>	42
<a href="#">content_formatting_factory.class.php</a>	43
<a href="#">Define BBCODE LANGUAGE</a>	43
<a href="#">Define DEFAULT LANGUAGE</a>	43
<a href="#">Define TINYMCE LANGUAGE</a>	43
<a href="#">parser.class.php</a>	44
<a href="#">Define ADD SLASHES</a>	44
<a href="#">Define DO NOT ADD SLASHES</a>	44
<a href="#">Define PARSER DO NOT STRIP SLASHES</a>	44
<a href="#">Define PARSER STRIP SLASHES</a>	44
<a href="#">Define PICK UP</a>	44
<a href="#">Define REIMPLANT</a>	44
<a href="#">template_highlighter.class.php</a>	45
<a href="#">Define TPL BRACES STYLE</a>	45
<a href="#">Define TPL KEYWORD STYLE</a>	45
<a href="#">Define TPL NESTED VARIABLE STYLE</a>	45
<a href="#">Define TPL SHARP STYLE</a>	45
<a href="#">Define TPL VARIABLE STYLE</a>	45
<a href="#">Class BBCodeHighlighter</a>	45
<a href="#">Constructor BBCodeHighlighter</a>	45
<a href="#">Method parse</a>	45
<a href="#">Class BBCodeParser</a>	46
<a href="#">Constructor BBCodeParser</a>	46
<a href="#">Method clear_html_br</a>	46
<a href="#">Method parse</a>	46
<a href="#">Class BBCodeUnparser</a>	46
<a href="#">Constructor BBCodeUnparser</a>	47
<a href="#">Method parse</a>	47
<a href="#">Class ContentFormattingFactory</a>	47
<a href="#">Var \$language_type</a>	47
<a href="#">Constructor ContentFormattingFactory</a>	47
<a href="#">Method get_available_tags</a>	48
<a href="#">Method get_editor</a>	48
<a href="#">Method get_language</a>	48
<a href="#">Method get_parser</a>	48

<a href="#">Method <u>get_second_parser</u></a>	48
<a href="#">Method <u>get_unparser</u></a>	48
<a href="#">Method <u>get_user_editor</u></a>	48
<a href="#">Method <u>set_language</u></a>	49
<a href="#">Class <u>ContentParser</u></a>	49
<a href="#">  Var <u>\$forbidden_tags</u></a>	49
<a href="#">  Var <u>\$html_auth</u></a>	50
<a href="#">  Var <u>\$tag</u></a>	50
<a href="#">  Constructor <u>ContentParser</u></a>	50
<a href="#">  Method <u>get_forbidden_tags</u></a>	50
<a href="#">  Method <u>parse</u></a>	50
<a href="#">  Method <u>set_forbidden_tags</u></a>	50
<a href="#">  Method <u>set_html_auth</u></a>	50
<a href="#">Class <u>ContentSecondParser</u></a>	51
<a href="#">  Constructor <u>ContentSecondParser</u></a>	51
<a href="#">  Method <u>export_html_text</u></a>	51
<a href="#">  Method <u>parse</u></a>	51
<a href="#">Class <u>ContentUnparser</u></a>	52
<a href="#">  Constructor <u>ContentUnparser</u></a>	52
<a href="#">Class <u>Parser</u></a>	52
<a href="#">  Var <u>\$array_tags</u></a>	52
<a href="#">  Var <u>\$content</u></a>	53
<a href="#">  Var <u>\$page_path</u></a>	53
<a href="#">  Var <u>\$path_to_root</u></a>	53
<a href="#">  Constructor <u>Parser</u></a>	53
<a href="#">  Method <u>get_content</u></a>	53
<a href="#">  Method <u>get_page_path</u></a>	53
<a href="#">  Method <u>get_path_to_root</u></a>	54
<a href="#">  Method <u>set_content</u></a>	54
<a href="#">  Method <u>set_page_path</u></a>	54
<a href="#">  Method <u>set_path_to_root</u></a>	54
<a href="#">Class <u>TemplateHighlighter</u></a>	55
<a href="#">  Constructor <u>TemplateHighlighter</u></a>	55
<a href="#">  Method <u>parse</u></a>	55
<a href="#">Class <u>TinyMCEParser</u></a>	55
<a href="#">  Constructor <u>TinyMCEParser</u></a>	56
<a href="#">  Method <u>parse</u></a>	56
<a href="#">Class <u>TinyMCEUnparser</u></a>	56
<a href="#">  Constructor <u>TinyMCEUnparser</u></a>	56
<a href="#">  Method <u>parse</u></a>	56
<a href="#">site_map.class.php</a>	57
<a href="#">  Define <u>SITE_MAP_AUTH_GUEST</u></a>	57
<a href="#">  Define <u>SITE_MAP_AUTH_USER</u></a>	57
<a href="#">  Define <u>SITE_MAP_FREQ_ALWAYS</u></a>	57
<a href="#">  Define <u>SITE_MAP_FREQ_DAILY</u></a>	57
<a href="#">  Define <u>SITE_MAP_FREQ_DEFAULT</u></a>	57
<a href="#">  Define <u>SITE_MAP_FREQ_HOURLY</u></a>	57
<a href="#">  Define <u>SITE_MAP_FREQ_MONTHLY</u></a>	57
<a href="#">  Define <u>SITE_MAP_FREQ_NEVER</u></a>	57

<a href="#">Define SITE_MAP_FREQ_WEEKLY</a>	57
<a href="#">Define SITE_MAP_FREQ_YEARLY</a>	57
<a href="#">Define SITE_MAP_PRIORITY_AVERAGE</a>	57
<a href="#">Define SITE_MAP_PRIORITY_HIGH</a>	57
<a href="#">Define SITE_MAP_PRIORITY_LOW</a>	57
<a href="#">Define SITE_MAP_PRIORITY_MAX</a>	57
<a href="#">Define SITE_MAP_PRIORITY_MIN</a>	57
<a href="#">Define SITE_MAP_SEARCH_ENGINE_MODE</a>	57
<a href="#">Define SITE_MAP_USER_MODE</a>	57
<a href="#">Class ModuleMap</a>	58
<a href="#">Var \$description</a>	58
<a href="#">Constructor ModuleMap</a>	58
<a href="#">Method export</a>	58
<a href="#">Method get_description</a>	59
<a href="#">Method set_description</a>	59
<a href="#">Class SiteMap</a>	59
<a href="#">Var \$elements</a>	60
<a href="#">Var \$site_name</a>	60
<a href="#">Constructor SiteMap</a>	60
<a href="#">Method add</a>	60
<a href="#">Method build_kernel_map</a>	60
<a href="#">Method build_modules_maps</a>	61
<a href="#">Method export</a>	61
<a href="#">Method get_site_name</a>	61
<a href="#">Method set_site_name</a>	61
<a href="#">Class SiteMapElement</a>	61
<a href="#">Var \$depth</a>	62
<a href="#">Constructor SiteMapElement</a>	62
<a href="#">Method export</a>	62
<a href="#">Method get_depth</a>	62
<a href="#">Method get_name</a>	63
<a href="#">Method set_depth</a>	63
<a href="#">Class SitemapExportConfig</a>	63
<a href="#">Var \$link_file</a>	63
<a href="#">Var \$module_map_file</a>	64
<a href="#">Var \$section_file</a>	64
<a href="#">Var \$site_map_file</a>	64
<a href="#">Constructor SitemapExportConfig</a>	64
<a href="#">Method get_link_stream</a>	64
<a href="#">Method get_module_map_stream</a>	65
<a href="#">Method get_section_stream</a>	65
<a href="#">Method get_site_map_stream</a>	65
<a href="#">Method set_link_stream</a>	65
<a href="#">Method set_module_map_stream</a>	65
<a href="#">Method set_section_stream</a>	65
<a href="#">Method set_site_map_stream</a>	65
<a href="#">Class SiteMapLink</a>	66
<a href="#">Var \$change_freq</a>	66
<a href="#">Var \$last_modification_date</a>	66

<a href="#">Var \$link</a>	67
<a href="#">Var \$name</a>	67
<a href="#">Var \$priority</a>	67
<a href="#">Method export</a>	67
<a href="#">Method get_change_freq</a>	68
<a href="#">Method get_last_modification_date</a>	68
<a href="#">Method get_link</a>	68
<a href="#">Method get_name</a>	68
<a href="#">Method get_priority</a>	68
<a href="#">Method get_url</a>	68
<a href="#">Method set_change_freq</a>	68
<a href="#">Method set_last_modification_date</a>	68
<a href="#">Method set_link</a>	69
<a href="#">Method set_name</a>	69
<a href="#">Method set_priority</a>	69
<a href="#">Method SitemapLink</a>	69
<a href="#">Class SiteMapSection</a>	70
<a href="#">Var \$elements</a>	70
<a href="#">Var \$link</a>	70
<a href="#">Constructor SiteMapSection</a>	70
<a href="#">Method add</a>	71
<a href="#">Method export</a>	71
<a href="#">Method get_link</a>	71
<a href="#">Method set_depth</a>	71
<a href="#">Method set_link</a>	72
<a href="#">atom.class.php</a>	73
<a href="#">Define DEFAULT_ATOM_TEMPLATE</a>	73
<a href="#">feed.class.php</a>	74
<a href="#">Define DEFAULT_FEED_NAME</a>	74
<a href="#">Define ERROR_GETTING_CACHE</a>	74
<a href="#">Define FEEDS_PATH</a>	74
<a href="#">feed_data.class.php</a>	75
<a href="#">Define FEED_DATA_CLASS</a>	75
<a href="#">rss.class.php</a>	76
<a href="#">Define DEFAULT_RSS_TEMPLATE</a>	76
<a href="#">Class ATOM</a>	76
<a href="#">Constructor ATOM</a>	76
<a href="#">Method load_file</a>	76
<a href="#">Class Feed</a>	77
<a href="#">Var \$data</a>	77
<a href="#">Var \$id_cat</a>	77
<a href="#">Var \$module_id</a>	77
<a href="#">Var \$name</a>	77
<a href="#">Var \$str</a>	77
<a href="#">Var \$tpl</a>	77
<a href="#">Constructor Feed</a>	77
<a href="#">Method cache</a>	78
<a href="#">Method clear_cache</a>	78
<a href="#">Method export</a>	78

<a href="#">Method get_cache_file_name</a>	78
<a href="#">Method get_feed_menu</a>	78
<a href="#">Method get_parsed</a>	79
<a href="#">Method is_in_cache</a>	79
<a href="#">Method load_data</a>	79
<a href="#">Method load_file</a>	79
<a href="#">Method read</a>	80
<a href="#">Method update_cache</a>	80
<a href="#">Class FeedData</a>	80
<a href="#">Var \$auth_bit</a>	81
<a href="#">Var \$date</a>	81
<a href="#">Var \$desc</a>	81
<a href="#">Var \$host</a>	81
<a href="#">Var \$items</a>	81
<a href="#">Var \$lang</a>	81
<a href="#">Var \$link</a>	81
<a href="#">Var \$title</a>	81
<a href="#">Constructor FeedData</a>	81
<a href="#">Method add_item</a>	81
<a href="#">Method get_date</a>	81
<a href="#">Method get_date_rfc822</a>	81
<a href="#">Method get_date_rfc3339</a>	82
<a href="#">Method get_desc</a>	82
<a href="#">Method get_host</a>	82
<a href="#">Method get_items</a>	82
<a href="#">Method get_lang</a>	82
<a href="#">Method get_link</a>	82
<a href="#">Method get_title</a>	82
<a href="#">Method serialize</a>	82
<a href="#">Method set_auth_bit</a>	82
<a href="#">Method set_date</a>	82
<a href="#">Method set_desc</a>	82
<a href="#">Method set_host</a>	82
<a href="#">Method set_lang</a>	83
<a href="#">Method set_link</a>	83
<a href="#">Method set_title</a>	83
<a href="#">Method subitems</a>	83
<a href="#">Class FeedItem</a>	83
<a href="#">Var \$auth</a>	84
<a href="#">Var \$date</a>	84
<a href="#">Var \$desc</a>	84
<a href="#">Var \$guid</a>	84
<a href="#">Var \$image_url</a>	84
<a href="#">Var \$link</a>	84
<a href="#">Var \$title</a>	84
<a href="#">Constructor FeedItem</a>	84
<a href="#">Method get_auth</a>	84
<a href="#">Method get_date</a>	84
<a href="#">Method get_date_rfc822</a>	84

<a href="#">Method <u>get_date</u> <u>rfc3339</u></a>	84
<a href="#">Method <u>get_desc</u></a>	84
<a href="#">Method <u>get_guid</u></a>	84
<a href="#">Method <u>get_image_url</u></a>	84
<a href="#">Method <u>get_link</u></a>	84
<a href="#">Method <u>get_title</u></a>	84
<a href="#">Method <u>set_auth</u></a>	84
<a href="#">Method <u>set_date</u></a>	85
<a href="#">Method <u>set_desc</u></a>	85
<a href="#">Method <u>set_guid</u></a>	85
<a href="#">Method <u>set_image_url</u></a>	85
<a href="#">Method <u>set_link</u></a>	85
<a href="#">Method <u>set_title</u></a>	86
<a href="#">Class <u>FeedsCat</u></a>	86
<a href="#">  Var <u>\$cat_name</u></a>	86
<a href="#">  Var <u>\$children</u></a>	86
<a href="#">  Var <u>\$id</u></a>	86
<a href="#">  Var <u>\$module_id</u></a>	87
<a href="#">  Constructor <u>FeedsCat</u></a>	87
<a href="#">  Method <u>add_child</u></a>	87
<a href="#">  Method <u>get_category_id</u></a>	87
<a href="#">  Method <u>get_category_name</u></a>	87
<a href="#">  Method <u>get_children</u></a>	87
<a href="#">  Method <u>get_module_id</u></a>	87
<a href="#">  Method <u>get_url</u></a>	87
<a href="#">Class <u>FeedsList</u></a>	88
<a href="#">  Var <u>\$list</u></a>	88
<a href="#">  Constructor <u>FeedsList</u></a>	88
<a href="#">  Method <u>add_feed</u></a>	88
<a href="#">  Method <u>get_feeds_list</u></a>	88
<a href="#">Class <u>RSS</u></a>	89
<a href="#">  Constructor <u>RSS</u></a>	89
<a href="#">  Method <u>load_file</u></a>	89
<a href="#">Package <u>core Procedural Elements</u></a>	91
<a href="#">  <u>application.class.php</u></a>	91
<a href="#">Define <u>APPLICATION_TYPE_KERNEL</u></a>	91
<a href="#">Define <u>APPLICATION_TYPE_MODULE</u></a>	91
<a href="#">Define <u>APPLICATION_TYPE_TEMPLATE</u></a>	91
<a href="#">  <u>cache.class.php</u></a>	92
<a href="#">Define <u>NO_FATAL_ERROR_CACHE</u></a>	92
<a href="#">Define <u>RELOAD_CACHE</u></a>	92
<a href="#">  <u>errors.class.php</u></a>	93
<a href="#">Define <u>ARCHIVE_ALL_ERRORS</u></a>	93
<a href="#">Define <u>ARCHIVE_ERROR</u></a>	93
<a href="#">Define <u>DISPLAY_ALL_ERROR</u></a>	93
<a href="#">Define <u>E_STRICT</u></a>	93
<a href="#">Define <u>NO_ARCHIVE_ERROR</u></a>	93
<a href="#">Define <u>NO_FILE_ERROR</u></a>	93
<a href="#">Define <u>NO_LINE_ERROR</u></a>	93

<a href="#">menu_service.class.php</a>	94
<a href="#">Define MOVE_DOWN</a>	94
<a href="#">Define MOVE_UP</a>	94
<a href="#">stats_saver.class.php</a>	95
<a href="#">updates.class.php</a>	96
<a href="#">Define CHECK_ALL_UPDATES</a>	96
<a href="#">Define CHECK_KERNEL</a>	96
<a href="#">Define CHECK_MODULES</a>	96
<a href="#">Define CHECK_THEMES</a>	96
<a href="#">Define PHPBOOST_OFFICIAL_REPOSITORY</a>	96
<a href="#">Define PHP_MIN_VERSION_UPDATES</a>	96
<b><a href="#">Package core Classes</a></b>	<b>97</b>
<a href="#">Class Application</a>	97
<a href="#">Var \$authors</a>	97
<a href="#">Var \$bug_corrections</a>	97
<a href="#">Var \$compatibility_max</a>	97
<a href="#">Var \$compatibility_min</a>	97
<a href="#">Var \$description</a>	97
<a href="#">Var \$download_url</a>	97
<a href="#">Var \$id</a>	97
<a href="#">Var \$improvements</a>	98
<a href="#">Var \$language</a>	98
<a href="#">Var \$localized_language</a>	98
<a href="#">Var \$name</a>	98
<a href="#">Var \$new_features</a>	98
<a href="#">Var \$priority</a>	98
<a href="#">Var \$pubdate</a>	98
<a href="#">Var \$repository</a>	98
<a href="#">Var \$security_improvements</a>	98
<a href="#">Var \$security_update</a>	98
<a href="#">Var \$type</a>	98
<a href="#">Var \$update_url</a>	98
<a href="#">Var \$version</a>	98
<a href="#">Var \$warning</a>	98
<a href="#">Var \$warning_level</a>	98
<a href="#">Constructor Application</a>	99
<a href="#">Method check_compatibility</a>	99
<a href="#">Method get_authors</a>	99
<a href="#">Method get_bug_corrections</a>	99
<a href="#">Method get_compatibility_max</a>	99
<a href="#">Method get_compatibility_min</a>	99
<a href="#">Method get_description</a>	99
<a href="#">Method get_download_url</a>	99
<a href="#">Method get_id</a>	99
<a href="#">Method get_identifier</a>	99
<a href="#">Method get_improvements</a>	99
<a href="#">Method get_language</a>	100
<a href="#">Method get_localized_language</a>	100
<a href="#">Method get_name</a>	100

<a href="#">Method get_new_features</a>	100
<a href="#">Method get_priority</a>	100
<a href="#">Method get_pubdate</a>	100
<a href="#">Method get_repository</a>	100
<a href="#">Method get_security_improvements</a>	100
<a href="#">Method get_security_update</a>	100
<a href="#">Method get_type</a>	100
<a href="#">Method get_update_url</a>	100
<a href="#">Method get_version</a>	100
<a href="#">Method get_warning</a>	101
<a href="#">Method get_warning_level</a>	101
<a href="#">Method load</a>	101
<a href="#">Class BreadCrumb</a>	101
<a href="#">Var \$array_links</a>	101
<a href="#">Constructor BreadCrumb</a>	102
<a href="#">Method add</a>	102
<a href="#">Method clean</a>	102
<a href="#">Method display</a>	102
<a href="#">Method remove_last</a>	102
<a href="#">Method reverse</a>	102
<a href="#">Class Cache</a>	102
<a href="#">Var \$files</a>	103
<a href="#">Constructor Cache</a>	103
<a href="#">Method delete_file</a>	103
<a href="#">Method generate_all_files</a>	103
<a href="#">Method generate_all_modules</a>	103
<a href="#">Method generate_file</a>	103
<a href="#">Method generate_module_file</a>	104
<a href="#">Method load</a>	104
<a href="#">Method write</a>	104
<a href="#">Class Errors</a>	104
<a href="#">Var \$archive_all</a>	105
<a href="#">Var \$personal_tpl</a>	105
<a href="#">Var \$redirect</a>	105
<a href="#">Var \$template</a>	105
<a href="#">Constructor Errors</a>	105
<a href="#">Method display</a>	105
<a href="#">Method get_erro_class</a>	105
<a href="#">Method get_last_error_log</a>	106
<a href="#">Method handler</a>	106
<a href="#">Method handler_php</a>	106
<a href="#">Method set_default_template</a>	106
<a href="#">Method set_template</a>	106
<a href="#">Class MenuService</a>	107
<a href="#">Method add_mini_menu</a>	107
<a href="#">Method add_mini_module</a>	107
<a href="#">Method assign_positions_conditions</a>	107
<a href="#">Method change_position</a>	108
<a href="#">Method delete</a>	108

<a href="#">Method delete_mini_menu</a>	108
<a href="#">Method delete_mini_module</a>	108
<a href="#">Method delete_module_feeds_menus</a>	109
<a href="#">Method disable</a>	109
<a href="#">Method enable</a>	109
<a href="#">Method enable_all</a>	109
<a href="#">Method generate_cache</a>	109
<a href="#">Method get_menus_map</a>	110
<a href="#">Method get_menu_list</a>	110
<a href="#">Method load</a>	110
<a href="#">Method move</a>	110
<a href="#">Method save</a>	110
<a href="#">Method str_to_location</a>	111
<a href="#">Method update_mini_menus_list</a>	111
<a href="#">Method update_mini_modules_list</a>	111
<a href="#">Method website_modules</a>	111
<a href="#">Class Repository</a>	111
<a href="#">Var \$url</a>	112
<a href="#">Var \$xml</a>	112
<a href="#">Constructor Repository</a>	112
<a href="#">Method check</a>	112
<a href="#">Method get_url</a>	112
<a href="#">Class StatsSaver</a>	112
<a href="#">Method compute_referer</a>	113
<a href="#">Method compute_users</a>	113
<a href="#">Class Updates</a>	113
<a href="#">Var \$apps</a>	113
<a href="#">Var \$repositories</a>	113
<a href="#">Constructor Updates</a>	113
<a href="#">Package db Procedural Elements</a>	115
<a href="#">mysql.class.php</a>	115
<a href="#">Define CONNECTED_TO_DATABASE</a>	115
<a href="#">Define CONNECTION_FAILED</a>	115
<a href="#">Define DBTYPE</a>	115
<a href="#">Define DB_NO_CONNECT</a>	115
<a href="#">Define ERRORS_MANAGEMENT_BY_RETURN</a>	115
<a href="#">Define EXPLICIT_ERRORS_MANAGEMENT</a>	115
<a href="#">Define LOW_PRIORITY</a>	115
<a href="#">Define UNEXISTING_DATABASE</a>	115
<a href="#">Package db Classes</a>	116
<a href="#">Class Backup</a>	116
<a href="#">Var \$backup_script</a>	116
<a href="#">Var \$tables</a>	116
<a href="#">Constructor Backup</a>	116
<a href="#">Method concatenate_to_query</a>	117
<a href="#">Method export_file</a>	117
<a href="#">Method extract_table_structure</a>	117
<a href="#">Method generate_create_table_query</a>	117

<a href="#">Method generate drop table query</a>	117
<a href="#">Method generate insert values query</a>	118
<a href="#">Method get script</a>	118
<a href="#">Method get tables list</a>	118
<a href="#">Method get tables number</a>	118
<a href="#">Method get tables properties list</a>	118
<a href="#">Method list db tables</a>	118
<a href="#">Class Sql</a>	118
<a href="#">  Var \$base_name</a>	119
<a href="#">  Var \$connected</a>	119
<a href="#">  Var \$link</a>	119
<a href="#">  Var \$req</a>	119
<a href="#">  Constructor Sql</a>	120
<a href="#">  Method affected_rows</a>	120
<a href="#">  Method auto_connect</a>	120
<a href="#">  Method clean_database_name</a>	120
<a href="#">  Method close</a>	120
<a href="#">  Method concat</a>	121
<a href="#">  Method connect</a>	121
<a href="#">  Method count_table</a>	121
<a href="#">  Method create_database</a>	122
<a href="#">  Method date_diff</a>	122
<a href="#">  Method drop_tables</a>	122
<a href="#">  Method escape</a>	122
<a href="#">  Method fetch_assoc</a>	122
<a href="#">  Method fetch_row</a>	123
<a href="#">  Method get_data_base_name</a>	123
<a href="#">  Method get_dbms_version</a>	123
<a href="#">  Method get_executed_requests_number</a>	123
<a href="#">  Method highlight_query</a>	123
<a href="#">  Method indent_query</a>	124
<a href="#">  Method insert_id</a>	124
<a href="#">  Method limit</a>	124
<a href="#">  Method list_databases</a>	124
<a href="#">  Method list_fields</a>	124
<a href="#">  Method list_tables</a>	125
<a href="#">  Method num_rows</a>	125
<a href="#">  Method optimize_tables</a>	125
<a href="#">  Method parse</a>	125
<a href="#">  Method query</a>	125
<a href="#">  Method query_array</a>	126
<a href="#">  Method query_close</a>	126
<a href="#">  Method query_inject</a>	126
<a href="#">  Method query_while</a>	127
<a href="#">  Method repair_tables</a>	127
<a href="#">  Method truncate_tables</a>	127
<a href="#">Package events Procedural Elements</a>	129
<a href="#">  administrator_alert.class.php</a>	129
<a href="#">  Define ADMIN_ALERT_HIGH_PRIORITY</a>	129

<a href="#">Define ADMIN_ALERT_LOW_PRIORITY</a>	129
<a href="#">Define ADMIN_ALERT_MEDIUM_PRIORITY</a>	129
<a href="#">Define ADMIN_ALERT_STATUS_PROCESSED</a>	129
<a href="#">Define ADMIN_ALERT_STATUS_UNREAD</a>	129
<a href="#">Define ADMIN_ALERT_VERY_HIGH_PRIORITY</a>	129
<a href="#">Define ADMIN_ALERT_VERY_LOW_PRIORITY</a>	129
<a href="#">administrator_alert_service.class.php</a>	130
<a href="#">Define ADMINISTRATOR_ALERT_TYPE</a>	130
<a href="#">contribution.class.php</a>	131
<a href="#">Define CONTRIBUTION_AUTH_BIT</a>	131
<a href="#">contribution_service.class.php</a>	132
<a href="#">Define CONTRIBUTION_TYPE</a>	132
<a href="#">event.class.php</a>	133
<a href="#">Define EVENT_STATUS_BEING_PROCESSED</a>	133
<a href="#">Define EVENT_STATUS_PROCESSED</a>	133
<a href="#">Define EVENT_STATUS_UNREAD</a>	133
<b><a href="#">Package events Classes</a></b>	134
<a href="#">Class AdministratorAlert</a>	134
<a href="#">Var \$priority</a>	134
<a href="#">Var \$properties</a>	134
<a href="#">Constructor AdministratorAlert</a>	135
<a href="#">Method build</a>	135
<a href="#">Method get_priority</a>	135
<a href="#">Method get_priority_name</a>	135
<a href="#">Method get_properties</a>	135
<a href="#">Method set_priority</a>	135
<a href="#">Method set_properties</a>	136
<a href="#">Class AdministratorAlertService</a>	136
<a href="#">Method compute_number_unread_alerts</a>	136
<a href="#">Method delete_alert</a>	136
<a href="#">Method find_by_criteria</a>	137
<a href="#">Method find_by_id</a>	137
<a href="#">Method find_by_identifier</a>	137
<a href="#">Method get_all_alerts</a>	138
<a href="#">Method get_number_alerts</a>	138
<a href="#">Method get_number_unread_alerts</a>	138
<a href="#">Method save_alert</a>	139
<a href="#">Class Contribution</a>	139
<a href="#">Var \$auth</a>	140
<a href="#">Var \$description</a>	140
<a href="#">Var \$fixer_id</a>	140
<a href="#">Var \$fixer_login</a>	140
<a href="#">Var \$fixing_date</a>	140
<a href="#">Var \$module</a>	141
<a href="#">Var \$poster_id</a>	141
<a href="#">Var \$poster_login</a>	141
<a href="#">Constructor Contribution</a>	141
<a href="#">Method build</a>	141
<a href="#">Method get_auth</a>	142

<a href="#">Method get description</a>	142
<a href="#">Method get fixer id</a>	142
<a href="#">Method get fixer login</a>	142
<a href="#">Method get fixing date</a>	142
<a href="#">Method get module</a>	142
<a href="#">Method get module name</a>	142
<a href="#">Method get poster id</a>	142
<a href="#">Method get poster login</a>	142
<a href="#">Method get status name</a>	143
<a href="#">Method set auth</a>	143
<a href="#">Method set description</a>	143
<a href="#">Method set fixer id</a>	143
<a href="#">Method set fixing date</a>	143
<a href="#">Method set module</a>	143
<a href="#">Method set poster id</a>	144
<a href="#">Method set status</a>	144
<a href="#">Class ContributionService</a>	144
<a href="#">Method compute number contrib for each profile</a>	145
<a href="#">Method delete contribution</a>	145
<a href="#">Method find by criteria</a>	145
<a href="#">Method find by id</a>	146
<a href="#">Method generate cache</a>	146
<a href="#">Method get all contributions</a>	146
<a href="#">Method save contribution</a>	147
<a href="#">Class Event</a>	147
<a href="#">Var \$creation_date</a>	147
<a href="#">Var \$current_status</a>	148
<a href="#">Var \$entitled</a>	148
<a href="#">Var \$fixing_url</a>	148
<a href="#">Var \$id</a>	148
<a href="#">Var \$identifier</a>	149
<a href="#">Var \$id in module</a>	149
<a href="#">Var \$must regenerate cache</a>	149
<a href="#">Var \$type</a>	149
<a href="#">Constructor Event</a>	149
<a href="#">Method build</a>	149
<a href="#">Method get creation_date</a>	150
<a href="#">Method get entitled</a>	150
<a href="#">Method get fixing_url</a>	150
<a href="#">Method get id</a>	150
<a href="#">Method get identifier</a>	150
<a href="#">Method get id in module</a>	150
<a href="#">Method get must regenerate cache</a>	150
<a href="#">Method get status</a>	150
<a href="#">Method get status_name</a>	151
<a href="#">Method get type</a>	151
<a href="#">Method set creation_date</a>	151
<a href="#">Method set entitled</a>	151
<a href="#">Method set fixing_url</a>	151

<a href="#">Method set_id</a>	151
<a href="#">Method set_identifier</a>	152
<a href="#">Method set_id_in_module</a>	152
<a href="#">Method set_must_regenerate_cache</a>	152
<a href="#">Method set_status</a>	152
<a href="#">Method set_type</a>	153
<b><a href="#">Package io_Procedural Elements</a></b>	<b>155</b>
<a href="#">mail.class.php</a>	155
<a href="#">Define CRLF</a>	155
<a href="#">Define MIME_FORMAT_HTML</a>	155
<a href="#">Define MIME_FORMAT_TEXT</a>	155
<a href="#">template.class.php</a>	156
<a href="#">Define AUTO_LOAD_FREQUENT_VARS</a>	156
<a href="#">Define DO_NOT_AUTO_LOAD_FREQUENT_VARS</a>	156
<a href="#">Define TEMPLATE_STRING_MODE</a>	156
<a href="#">Define TEMPLATE_WRITE_MODE</a>	156
<a href="#">upload.class.php</a>	157
<a href="#">Define CHECK_EXIST</a>	157
<a href="#">Define DELETE_ON_ERROR</a>	157
<a href="#">Define NO_DELETE_ON_ERROR</a>	157
<a href="#">Define NO_UNIQ_NAME</a>	157
<a href="#">Define UNIQ_NAME</a>	157
<b><a href="#">Package io Classes</a></b>	<b>158</b>
<a href="#">Class Mail</a>	158
<a href="#">Var \$content</a>	158
<a href="#">Var \$format</a>	158
<a href="#">Var \$headers</a>	159
<a href="#">Var \$object</a>	159
<a href="#">Var \$recipients</a>	159
<a href="#">Var \$sender_mail</a>	159
<a href="#">Var \$sender_name</a>	159
<a href="#">Constructor Mail</a>	160
<a href="#">Method check_validity</a>	160
<a href="#">Method get_content</a>	160
<a href="#">Method get_headers</a>	160
<a href="#">Method get_mime</a>	160
<a href="#">Method get_object</a>	160
<a href="#">Method get_recipients</a>	160
<a href="#">Method get_sender_mail</a>	160
<a href="#">Method get_sender_name</a>	160
<a href="#">Method send</a>	161
<a href="#">Method send_from_properties</a>	161
<a href="#">Method set_content</a>	161
<a href="#">Method set_headers</a>	161
<a href="#">Method set_mime</a>	162
<a href="#">Method set_object</a>	162
<a href="#">Method set_recipients</a>	162
<a href="#">Method set_sender</a>	162

<a href="#">Method generate headers</a>	162
<a href="#">Class Template</a>	163
<a href="#">Var \$files</a>	164
<a href="#">Var \$module_data_path</a>	164
<a href="#">Var \$return_mode</a>	164
<a href="#">Var \$template</a>	165
<a href="#">Var \$tpl</a>	165
<a href="#">Constructor Template</a>	165
<a href="#">Method assign_block_vars</a>	165
<a href="#">Method assign_vars</a>	166
<a href="#">Method copy</a>	166
<a href="#">Method get_module_data_path</a>	166
<a href="#">Method parse</a>	166
<a href="#">Method pparse</a>	166
<a href="#">Method set_filenames</a>	167
<a href="#">Method unassign_block_vars</a>	167
<a href="#">Method check_cache_file</a>	167
<a href="#">Method check_file</a>	168
<a href="#">Method include</a>	168
<a href="#">Method load</a>	168
<a href="#">Class Upload</a>	169
<a href="#">Var \$base_directory</a>	169
<a href="#">Var \$error</a>	169
<a href="#">Var \$extension</a>	169
<a href="#">Var \$filename</a>	169
<a href="#">Constructor Upload</a>	169
<a href="#">Method file</a>	170
<a href="#">Method validate_img</a>	170
<a href="#">file.class.php</a>	171
<a href="#">Define ADD</a>	171
<a href="#">Define CLOSEFILE</a>	171
<a href="#">Define ERASE</a>	171
<a href="#">Define LOCK</a>	171
<a href="#">Define NOTCLOSEFILE</a>	171
<a href="#">Define READ</a>	171
<a href="#">Define READ_WRITE</a>	171
<a href="#">Define WRITE</a>	171
<a href="#">file_system_element.class.php</a>	172
<a href="#">Define OPEN_AFTER</a>	172
<a href="#">Define OPEN_NOW</a>	172
<a href="#">Class File</a>	172
<a href="#">Var \$contents</a>	172
<a href="#">Var \$fd</a>	172
<a href="#">Var \$lines</a>	173
<a href="#">Var \$mode</a>	173
<a href="#">Constructor File</a>	173
<a href="#">Method close</a>	173
<a href="#">Method delete</a>	173
<a href="#">Method finclude</a>	173

<a href="#">Method flush</a>	174
<a href="#">Method frequire</a>	174
<a href="#">Method get_contents</a>	174
<a href="#">Method get_last_access_date</a>	174
<a href="#">Method get_last_modification_date</a>	174
<a href="#">Method get_lines</a>	174
<a href="#">Method is_open</a>	175
<a href="#">Method lock</a>	175
<a href="#">Method open</a>	175
<a href="#">Method unlock</a>	175
<a href="#">Method write</a>	175
<a href="#">Class FileSystemElement</a>	175
<a href="#">Var \$is_open</a>	176
<a href="#">Var \$path</a>	176
<a href="#">Constructor FileSystemElement</a>	176
<a href="#">Method change_chmod</a>	176
<a href="#">Method delete</a>	177
<a href="#">Method exists</a>	177
<a href="#">Method get</a>	177
<a href="#">Method get_name</a>	177
<a href="#">Method open</a>	177
<a href="#">Method write</a>	177
<a href="#">Class Folder</a>	178
<a href="#">Var \$files</a>	178
<a href="#">Var \$folders</a>	178
<a href="#">Constructor Folder</a>	178
<a href="#">Method delete</a>	178
<a href="#">Method get_all_content</a>	179
<a href="#">Method get_files</a>	179
<a href="#">Method get_first_folder</a>	179
<a href="#">Method get_folders</a>	179
<a href="#">Method open</a>	179
<a href="#">Package members Procedural Elements</a>	181
<a href="#">authorizations.class.php</a>	181
<a href="#">Define AUTH_CHILD_PRIORITY</a>	181
<a href="#">Define AUTH_PARENT_PRIORITY</a>	181
<a href="#">groups.class.php</a>	182
<a href="#">Define ADMIN_NOAUTH_DEFAULT</a>	182
<a href="#">Define GROUP_DEFAULT_IDSELECT</a>	182
<a href="#">Define GROUP_DISABLED_ADVANCED_AUTH</a>	182
<a href="#">Define GROUP_DISABLE_SELECT</a>	182
<a href="#">pm.class.php</a>	183
<a href="#">Define CHECK_PM_BOX</a>	183
<a href="#">Define DEL_PM_CONVERS</a>	183
<a href="#">Define NOCHECK_PM_BOX</a>	183
<a href="#">Define SYSTEM_PM</a>	183
<a href="#">Define UPDATE_MBR_PM</a>	183
<a href="#">session.class.php</a>	184
<a href="#">Define ALREADY_HASHED</a>	184

<a href="#">Define AUTOCONNECT</a>	184
<a href="#">Define NO_AUTOCONNECT</a>	184
<a href="#">Define SEASURF_ATTACK_ERROR_PAGE</a>	184
<a href="#">uploads.class.php</a>	185
<a href="#">Define ADMIN_NO_CHECK</a>	185
<a href="#">Define EMPTY_FOLDER</a>	185
<a href="#">user.class.php</a>	186
<a href="#">Define GROUP_TYPE</a>	186
<a href="#">Define RANK_TYPE</a>	186
<a href="#">Define USER_TYPE</a>	186
<b><a href="#">Package members Classes</a></b>	187
<a href="#">Class Authorizations</a>	187
<a href="#">Method auth_array_simple</a>	187
<a href="#">Method build_auth_array_from_form</a>	188
<a href="#">Method capture_and_shift_bit_auth</a>	188
<a href="#">Method check_auth</a>	188
<a href="#">Method generate_select</a>	189
<a href="#">Method merge_auth</a>	189
<a href="#">Class Group</a>	190
<a href="#">Var \$groups_auth</a>	190
<a href="#">Var \$groups_name</a>	190
<a href="#">Constructor Group</a>	190
<a href="#">Method add_member</a>	190
<a href="#">Method edit_member</a>	190
<a href="#">Method get_groups_array</a>	191
<a href="#">Method remove_member</a>	191
<a href="#">Class PrivateMsg</a>	191
<a href="#">Var \$pm_convers_id</a>	191
<a href="#">Var \$pm_msg_id</a>	191
<a href="#">Method count_conversations</a>	191
<a href="#">Method delete</a>	192
<a href="#">Method delete_conversation</a>	192
<a href="#">Method send</a>	192
<a href="#">Method start_conversation</a>	193
<a href="#">Class Session</a>	193
<a href="#">Var \$autoconnect</a>	193
<a href="#">Var \$data</a>	193
<a href="#">Var \$session_mod</a>	193
<a href="#">Method act</a>	193
<a href="#">Method check</a>	193
<a href="#">Method csrf_get_protect</a>	194
<a href="#">Method csrf_post_protect</a>	194
<a href="#">Method end</a>	194
<a href="#">Method garbage_collector</a>	194
<a href="#">Method get_module_parameters</a>	195
<a href="#">Method get_token</a>	195
<a href="#">Method load</a>	195
<a href="#">Method set_module_parameters</a>	195
<a href="#">Method start</a>	195

<a href="#">Class Uploads</a>	196
<a href="#">Var \$base_directory</a>	196
<a href="#">Var \$error</a>	196
<a href="#">Var \$extension</a>	196
<a href="#">Var \$filename</a>	196
<a href="#">Method Add_folder</a>	196
<a href="#">Method Del_file</a>	196
<a href="#">Method Del_folder</a>	196
<a href="#">Method Empty_folder_member</a>	197
<a href="#">Method Find_subfolder</a>	197
<a href="#">Method get_admin_url</a>	197
<a href="#">Method get_img_mimetype</a>	197
<a href="#">Method get_url</a>	197
<a href="#">Method Member_memory_used</a>	197
<a href="#">Method Move_file</a>	198
<a href="#">Method Move_folder</a>	198
<a href="#">Method Rename_file</a>	198
<a href="#">Method Rename_folder</a>	198
<a href="#">Class User</a>	199
<a href="#">Var \$groups_auth</a>	199
<a href="#">Var \$user_data</a>	199
<a href="#">Var \$user_groups</a>	199
<a href="#">Constructor User</a>	199
<a href="#">Method check_auth</a>	199
<a href="#">Method check_level</a>	200
<a href="#">Method check_max_value</a>	200
<a href="#">Method get_attribute</a>	200
<a href="#">Method get_groups</a>	200
<a href="#">Method get_group_color</a>	200
<a href="#">Method get_id</a>	201
<a href="#">Method set_user_lang</a>	201
<a href="#">Method set_user_theme</a>	201
<a href="#">Method update_user_lang</a>	201
<a href="#">Method update_user_theme</a>	201
<a href="#">Package menu Procedural Elements</a>	204
<a href="#">menu.class.php</a>	204
<a href="#">Define BLOCK_POSITION_ALL</a>	204
<a href="#">Define BLOCK_POSITION_BOTTOM_CENTRAL</a>	204
<a href="#">Define BLOCK_POSITION_FOOTER</a>	204
<a href="#">Define BLOCK_POSITION_HEADER</a>	204
<a href="#">Define BLOCK_POSITION_LEFT</a>	204
<a href="#">Define BLOCK_POSITION_NOT_ENABLED</a>	204
<a href="#">Define BLOCK_POSITION_RIGHT</a>	204
<a href="#">Define BLOCK_POSITION_SUB_HEADER</a>	204
<a href="#">Define BLOCK_POSITION_TOP_CENTRAL</a>	204
<a href="#">Define BLOCK_POSITION_TOP_FOOTER</a>	204
<a href="#">Define MENU_AUTH_BIT</a>	204
<a href="#">Define MENU_ENABLED</a>	204
<a href="#">Define MENU_ENABLE_OR_NOT</a>	204

<a href="#">Define MENU NOT_ENABLED</a>	204
<a href="#">Define MENU CLASS</a>	204
<b><a href="#">Package menu Classes</a></b>	<b>205</b>
<a href="#">Class Menu</a>	205
<a href="#">Var \$auth</a>	205
<a href="#">Var \$block</a>	205
<a href="#">Var \$enabled</a>	206
<a href="#">Var \$id</a>	206
<a href="#">Var \$position</a>	206
<a href="#">Var \$title</a>	206
<a href="#">Constructor Menu</a>	206
<a href="#">Method admin_display</a>	207
<a href="#">Method cache_export</a>	207
<a href="#">Method cache_export_begin</a>	207
<a href="#">Method cache_export_end</a>	207
<a href="#">Method display</a>	207
<a href="#">Method enabled</a>	207
<a href="#">Method get_auth</a>	207
<a href="#">Method get_block</a>	207
<a href="#">Method get_block_position</a>	207
<a href="#">Method get_id</a>	207
<a href="#">Method get_title</a>	207
<a href="#">Method id</a>	208
<a href="#">Method is_enabled</a>	208
<a href="#">Method set_auth</a>	208
<a href="#">Method set_block</a>	208
<a href="#">Method set_block_position</a>	208
<a href="#">Method set_title</a>	208
<a href="#">Method assign</a>	208
<a href="#">content.menu.class.php</a>	210
<a href="#">Define CONTENT_MENU CLASS</a>	210
<a href="#">Class ContentMenu</a>	210
<a href="#">Var \$content</a>	210
<a href="#">Var \$display_title</a>	210
<a href="#">Constructor ContentMenu</a>	210
<a href="#">Method cache_export</a>	211
<a href="#">Method display</a>	211
<a href="#">Method get_content</a>	211
<a href="#">Method get_display_title</a>	211
<a href="#">Method set_content</a>	211
<a href="#">Method set_display_title</a>	211
<a href="#">feed.menu.class.php</a>	212
<a href="#">Define FEED_MENU CLASS</a>	212
<a href="#">Class FeedMenu</a>	212
<a href="#">Var \$begin_at</a>	212
<a href="#">Var \$category</a>	212
<a href="#">Var \$module_id</a>	212
<a href="#">Var \$name</a>	212
<a href="#">Var \$number</a>	212

<a href="#">Var \$url</a>	212
<a href="#">Constructor FeedMenu</a>	213
<a href="#">Method cache_export</a>	213
<a href="#">Method display</a>	213
<a href="#">Method get_module_id</a>	213
<a href="#">Method get_template</a>	213
<a href="#">Method get_url</a>	213
<a href="#">Method set_cat</a>	213
<a href="#">Method set_module_id</a>	214
<a href="#">Method set_name</a>	214
<a href="#">links_menu.class.php</a>	215
<a href="#">Define HORIZONTAL_MENU</a>	215
<a href="#">Define HORIZONTAL_SCROLLING_MENU</a>	215
<a href="#">Define LINKS_MENU_CLASS</a>	215
<a href="#">Define TREE_MENU</a>	215
<a href="#">Define VERTICAL_MENU</a>	215
<a href="#">Define VERTICAL_SCROLLING_MENU</a>	215
<a href="#">links_menu_element.class.php</a>	216
<a href="#">Define LINKS_MENU_ELEMENT_CLASS</a>	216
<a href="#">Define LINKS_MENU_ELEMENT_CLASSIC_DISPLAYING</a>	216
<a href="#">Define LINKS_MENU_ELEMENT_FULL_DISPLAYING</a>	216
<a href="#">links_menu_link.class.php</a>	217
<a href="#">Define LINKS_MENU_LINK_CLASS</a>	217
<a href="#">Class LinksMenu</a>	217
<a href="#">Var \$elements</a>	217
<a href="#">Var \$type</a>	217
<a href="#">Constructor LinksMenu</a>	218
<a href="#">Method add</a>	218
<a href="#">Method add_array</a>	218
<a href="#">Method cache_export</a>	218
<a href="#">Method display</a>	218
<a href="#">Method get_children</a>	219
<a href="#">Method get_menu_types_list</a>	219
<a href="#">Method get_type</a>	219
<a href="#">Method set_type</a>	219
<a href="#">Method update_uid</a>	219
<a href="#">Method parent</a>	219
<a href="#">Class LinksMenuElement</a>	220
<a href="#">Var \$depth</a>	220
<a href="#">Var \$image</a>	220
<a href="#">Var \$uid</a>	221
<a href="#">Var \$url</a>	221
<a href="#">Constructor LinksMenuElement</a>	221
<a href="#">Method cache_export</a>	221
<a href="#">Method cache_export_begin</a>	221
<a href="#">Method cache_export_end</a>	221
<a href="#">Method display</a>	221
<a href="#">Method get_image</a>	222
<a href="#">Method get_uid</a>	222

<a href="#">Method <code>get_url</code></a>	222
<a href="#">Method <code>set_image</code></a>	222
<a href="#">Method <code>set_url</code></a>	222
<a href="#">Method <code>update_uid</code></a>	223
<a href="#">Method <code>assign</code></a>	223
<a href="#">Method <code>parent</code></a>	223
<a href="#">Class <code>LinksMenuLink</code></a>	223
<a href="#">Constructor <code>LinksMenuLink</code></a>	224
<a href="#">Method <code>cache_export</code></a>	224
<a href="#">Method <code>display</code></a>	224
<a href="#">mini_menu.class.php</a>	225
<a href="#">Define <code>MINI_MENU_CLASS</code></a>	225
<a href="#">Class <code>MiniMenu</code></a>	225
<a href="#">Var <code>\$function_name</code></a>	225
<a href="#">Constructor <code>MiniMenu</code></a>	225
<a href="#">Method <code>cache_export</code></a>	225
<a href="#">module_mini_menu.class.php</a>	226
<a href="#">Define <code>MODULE_MINI_MENU_CLASS</code></a>	226
<a href="#">Class <code>ModuleMiniMenu</code></a>	226
<a href="#">Var <code>\$filename</code></a>	226
<a href="#">Constructor <code>ModuleMiniMenu</code></a>	226
<a href="#">Method <code>cache_export</code></a>	226
<a href="#">Method <code>get_title</code></a>	226
<b><a href="#">Package modules Procedural Elements</a></b>	228
<a href="#">module_interface.class.php</a>	228
<a href="#">Define <code>ACCES_DENIED</code></a>	228
<a href="#">Define <code>FUNCTIONNALITY_NOT_IMPLEMENTED</code></a>	228
<a href="#">Define <code>MODULE_ATTRIBUTE_DOES_NOT_EXIST</code></a>	228
<a href="#">Define <code>MODULE_NOT_AVAILABLE</code></a>	228
<a href="#">Define <code>MODULE_NOT_YET_IMPLEMENTED</code></a>	228
<a href="#">packages_manager.class.php</a>	229
<a href="#">Define <code>CONFIG_CONFLICT</code></a>	229
<a href="#">Define <code>DO_NOT_GENERATE_CACHE_AFTER_THE_OPERATION</code></a>	229
<a href="#">Define <code>GENERATE_CACHE_AFTER_THE_OPERATION</code></a>	229
<a href="#">Define <code>MODULE_ALREADY_INSTALLED</code></a>	229
<a href="#">Define <code>MODULE_FILES_COULD_NOT_BE_DROPPED</code></a>	229
<a href="#">Define <code>MODULE_INSTALLED</code></a>	229
<a href="#">Define <code>MODULE_UNINSTALLED</code></a>	229
<a href="#">Define <code>NOT_INSTALLED_MODULE</code></a>	229
<a href="#">Define <code>PHP_VERSION_CONFLICT</code></a>	229
<a href="#">Define <code>UNEXISTING_MODULE</code></a>	229
<b><a href="#">Package modules Classes</a></b>	230
<a href="#">Class <code>ModuleInterface</code></a>	230
<a href="#">Var <code>\$attributes</code></a>	230
<a href="#">Var <code>\$enabled</code></a>	230
<a href="#">Var <code>\$errors</code></a>	230
<a href="#">Var <code>\$functionalities</code></a>	231
<a href="#">Var <code>\$id</code></a>	231

<a href="#">Var \$infos</a>	231
<a href="#">Var \$name</a>	231
<a href="#">Constructor ModuleInterface</a>	232
<a href="#">Method functionality</a>	232
<a href="#">Method get_attribute</a>	232
<a href="#">Method get_errors</a>	232
<a href="#">Method get_id</a>	232
<a href="#">Method get_infos</a>	232
<a href="#">Method get_name</a>	232
<a href="#">Method got_error</a>	232
<a href="#">Method has_functionalities</a>	232
<a href="#">Method has_functionality</a>	233
<a href="#">Method is_enabled</a>	233
<a href="#">Method set_attribute</a>	233
<a href="#">Method set_error</a>	233
<a href="#">Method unset_attribute</a>	233
<a href="#">Class ModulesDiscoveryService</a>	234
<a href="#">Var \$available_modules</a>	234
<a href="#">Var \$loaded_modules</a>	234
<a href="#">Constructor ModulesDiscoveryService</a>	234
<a href="#">Method functionality</a>	234
<a href="#">Method get_all_modules</a>	234
<a href="#">Method get_available_modules</a>	234
<a href="#">Method get_module</a>	235
<a href="#">Class PackagesManager</a>	235
<a href="#">Method install_module</a>	235
<a href="#">Method uninstall_module</a>	236
<a href="#">Package util Procedural Elements</a>	238
<a href="#">captcha.class.php</a>	238
<a href="#">Define CAPTCHA_EASY</a>	238
<a href="#">Define CAPTCHA_HARD</a>	238
<a href="#">Define CAPTCHA_NORMAL</a>	238
<a href="#">Define CAPTCHA_VERY_EASY</a>	238
<a href="#">Define CAPTCHA_VERY_HARD</a>	238
<a href="#">date.class.php</a>	239
<a href="#">Define DATE_FORMAT</a>	239
<a href="#">Define DATE_FORMAT_LONG</a>	239
<a href="#">Define DATE_FORMAT_SHORT</a>	239
<a href="#">Define DATE_FORMAT_TINY</a>	239
<a href="#">Define DATE_FROM_STRING</a>	239
<a href="#">Define DATE_NOW</a>	239
<a href="#">Define DATE_RFC822_F</a>	239
<a href="#">Define DATE_RFC822_FORMAT</a>	239
<a href="#">Define DATE_RFC3339_F</a>	239
<a href="#">Define DATE_RFC3339_FORMAT</a>	239
<a href="#">Define DATE_TIMESTAMP</a>	239
<a href="#">Define DATE_YEAR_MONTH_DAY</a>	239
<a href="#">Define DATE_YEAR_MONTH_DAY_HOUR_MINUTE_SECOND</a>	239
<a href="#">Define TIMEZONE_AUTO</a>	239

<a href="#">images_stats.class.php</a>	240
<a href="#">Define DRAW_LEGEND</a>	240
<a href="#">Define DRAW_VALUES</a>	240
<a href="#">Define FRANKLINBC_TTF</a>	240
<a href="#">Define NO_ALLOCATE_COLOR</a>	240
<a href="#">Define NO_DRAW_LEGEND</a>	240
<a href="#">Define NO_DRAW_PERCENT</a>	240
<a href="#">Define NO_DRAW_VALUES</a>	240
<a href="#">pagination.class.php</a>	241
<a href="#">Define LINK_START_PAGE</a>	241
<a href="#">Define NO_PREVIOUS_NEXT_LINKS</a>	241
<a href="#">unusual_functions.inc.php</a>	242
<a href="#">Function get_server_url_page</a>	242
<a href="#">Function securit_register_globals</a>	242
<a href="#">url.class.php</a>	243
<a href="#">Define SERVER_URL</a>	243
<a href="#">Define URL_CLASS</a>	243
<b>Package util Classes</b>	244
<a href="#">Class Bench</a>	244
<a href="#">Var \$duration</a>	244
<a href="#">Var \$start</a>	244
<a href="#">Constructor Bench</a>	245
<a href="#">Method get_microtime</a>	245
<a href="#">Method stop</a>	245
<a href="#">Method to_string</a>	245
<a href="#">Class Captcha</a>	245
<a href="#">Var \$code</a>	245
<a href="#">Var \$difficulty</a>	246
<a href="#">Var \$font</a>	246
<a href="#">Var \$gd_loaded</a>	246
<a href="#">Var \$height</a>	246
<a href="#">Var \$instance</a>	246
<a href="#">Var \$width</a>	246
<a href="#">Constructor Captcha</a>	246
<a href="#">Method display</a>	246
<a href="#">Method display_form</a>	246
<a href="#">Method is_available</a>	246
<a href="#">Method is_valid</a>	246
<a href="#">Method js_require</a>	246
<a href="#">Method set_difficulty</a>	247
<a href="#">Method set_font</a>	247
<a href="#">Method set_height</a>	247
<a href="#">Method set_instance</a>	247
<a href="#">Method set_width</a>	247
<a href="#">Method update_instance</a>	248
<a href="#">Class Date</a>	248
<a href="#">Var \$timestamp</a>	248
<a href="#">Constructor Date</a>	248
<a href="#">Method check_date</a>	249

<a href="#">Method format</a>	250
<a href="#">Method get_day</a>	250
<a href="#">Method get_hours</a>	250
<a href="#">Method get_minutes</a>	250
<a href="#">Method get_month</a>	250
<a href="#">Method get_seconds</a>	251
<a href="#">Method get_timestamp</a>	251
<a href="#">Method get_year</a>	251
<a href="#">Method to_date</a>	251
<a href="#">Class MiniCalendar</a>	251
<a href="#">Var \$date</a>	251
<a href="#">Var \$form_name</a>	251
<a href="#">Var \$num_instance</a>	252
<a href="#">Var \$style</a>	252
<a href="#">Constructor MiniCalendar</a>	252
<a href="#">Method display</a>	252
<a href="#">Method get_date</a>	252
<a href="#">Method retrieve_date</a>	252
<a href="#">Method set_date</a>	253
<a href="#">Method set_style</a>	253
<a href="#">Class Pagination</a>	253
<a href="#">Var \$nbr_end_links</a>	254
<a href="#">Var \$nbr_start_links</a>	254
<a href="#">Var \$page</a>	254
<a href="#">Var \$var_page</a>	254
<a href="#">Constructor Pagination</a>	254
<a href="#">Method display</a>	254
<a href="#">Method get_current_page</a>	254
<a href="#">Method get_first_msg</a>	254
<a href="#">Class Stats</a>	255
<a href="#">Var \$array_allocated_color</a>	255
<a href="#">Var \$array_color_stats</a>	255
<a href="#">Var \$color_index</a>	255
<a href="#">Var \$data_stats</a>	255
<a href="#">Var \$decimal</a>	256
<a href="#">Var \$nbr_color</a>	256
<a href="#">Var \$nbr_entry</a>	256
<a href="#">Constructor Stats</a>	256
<a href="#">Method draw_ellipse</a>	256
<a href="#">Method draw_graph</a>	256
<a href="#">Method draw_histogram</a>	256
<a href="#">Method load_data</a>	256
<a href="#">Class Url</a>	257
<a href="#">Var \$is_relative</a>	257
<a href="#">Var \$path_to_root</a>	257
<a href="#">Var \$server_url</a>	257
<a href="#">Var \$url</a>	257
<a href="#">Constructor Url</a>	257
<a href="#">Method absolute</a>	258

<a href="#">Method check_wellformness</a>	258
<a href="#">Method compress</a>	259
<a href="#">Method get_absolute_root</a>	259
<a href="#">Method get_relative</a>	259
<a href="#">Method get_wellformness_regex</a>	259
<a href="#">Method html_convert_absolute2root_relative</a>	260
<a href="#">Method html_convert_root_relative2absolute</a>	260
<a href="#">Method html_convert_root_relative2relative</a>	261
<a href="#">Method is_relative</a>	261
<a href="#">Method path_to_root</a>	261
<a href="#">Method relative</a>	262
<a href="#">Method root_to_local</a>	262
<a href="#">Method server_url</a>	262
<b><a href="#">Package phpboost Procedural Elements</a></b>	264
<a href="#">functions.inc.php</a>	264
<a href="#">Define ADDSLASHES_AUTO</a>	264
<a href="#">Define ADDSLASHES_FORCE</a>	264
<a href="#">Define ADDSLASHES_NONE</a>	264
<a href="#">Function array_combine</a>	264
<a href="#">Function check_mail</a>	264
<a href="#">Function check_nbr_links</a>	264
<a href="#">Define CLASS_IMPORT</a>	265
<a href="#">Function delete_file</a>	265
<a href="#">Function display_comments</a>	265
<a href="#">Function display_editor</a>	265
<a href="#">Function file_get_contents_emulate</a>	266
<a href="#">Function find_require_dir</a>	266
<a href="#">Function get_ini_config</a>	266
<a href="#">Function get_module_name</a>	266
<a href="#">Function get_start_page</a>	267
<a href="#">Function get_uid</a>	267
<a href="#">Function get_ulang</a>	267
<a href="#">Function get_utheme</a>	267
<a href="#">Function gmdate_format</a>	267
<a href="#">Function htmlspecialchars_decode</a>	267
<a href="#">Function html_entity_decode</a>	267
<a href="#">Define HTML_NO_PROTECT</a>	268
<a href="#">Define HTML_PROTECT</a>	268
<a href="#">Function import</a>	268
<a href="#">Function inc</a>	268
<a href="#">Define INC_IMPORT</a>	268
<a href="#">Define LIB_IMPORT</a>	268
<a href="#">Function load_ini_file</a>	268
<a href="#">Function load_menu_lang</a>	268
<a href="#">Function load_module_lang</a>	269
<a href="#">Define MAGIC_QUOTES_DISABLED</a>	269
<a href="#">Define NO_EDITOR_UNPARSE</a>	269
<a href="#">Define NO_FATAL_ERROR</a>	269
<a href="#">Define NO_UPDATE_PAGES</a>	269

<a href="#">Function number_round</a>	269
<a href="#">Function numeric</a>	269
<a href="#">Function of_class</a>	270
<a href="#">Function pages_displayed</a>	270
<a href="#">Function parse_ini_array</a>	270
<a href="#">Function phpboost_version</a>	270
<a href="#">Function redirect</a>	270
<a href="#">Function redirect_confirm</a>	271
<a href="#">Function req</a>	271
<a href="#">Function retrieve</a>	271
<a href="#">Function second_parse</a>	272
<a href="#">Function second_parse_url</a>	272
<a href="#">Function set_subregex_multiplicity</a>	273
<a href="#">Function strhash</a>	273
<a href="#">Function strparse</a>	273
<a href="#">Function strprotect</a>	274
<a href="#">Function strtodate</a>	274
<a href="#">Function strtotimestamp</a>	275
<a href="#">Function substr_html</a>	275
<a href="#">Define TIMEZONE_SITE</a>	275
<a href="#">Define TIMEZONE_SYSTEM</a>	275
<a href="#">Define TIMEZONE_USER</a>	275
<a href="#">Function to_js_string</a>	275
<a href="#">Function unparse</a>	276
<a href="#">Function url</a>	276
<a href="#">Function url_encode_rewrite</a>	276
<a href="#">Function wordwrap_html</a>	277
<a href="#">Appendices</a>	278
<a href="#">Appendix A - Class Trees</a>	279
<a href="#">members</a>	279
<a href="#">phpboost</a>	279
<a href="#">core</a>	280
<a href="#">menu</a>	281
<a href="#">db</a>	281
<a href="#">builder</a>	281
<a href="#">modules</a>	282
<a href="#">content</a>	283
<a href="#">events</a>	285
<a href="#">io</a>	286
<a href="#">util</a>	287



# Package builder Procedural Elements

## form\_builder.class.php

This class allows you to manage easily the forms in your modules. A lot of fields and options are supported, for further details refer yourself to each field class.

Example of use :

```
1     import('builder/form/form_builder');
2     $form = new FormBuilder('test', '');
3
4     //##### First fieldset #####//
5     $fieldset = new FormFieldset('Form Test');
6
7     $fieldset-> add_field(new FormTextEdit('login', array('title' => 'Login', 'subtitle'
=> 'Enter your login', 'class' => 'text', 'required' => true, 'required_alert' => 'Login
field has to be filled')));
8     //Textarea field
9     $fieldset-> add_field(new FormTextarea('contents', array('title' => 'Description',
'subtitle' => 'Enter a description', 'rows' => 10, 'cols' => 10, 'required' => true,
'required_alert' => 'Content field has to be filled')));
10    $fieldset-> add_field(new FormTextarea('comments', array('title' => 'Comments',
'subtitle' => '', 'rows' => 4, 'cols' => 5, 'editor' => false)));
11    //Radio button field
12    $fieldset-> add_field(new FormRadioChoice('choice', array('title' => 'Answer'),
13        new FormRadioChoiceOption(array('optiontitle' => 'Choix1', 'value' => 1)),
14        new FormRadioChoiceOption(array('optiontitle' => 'Choix2', 'value' => 2,
'subtitle' => 'checked' => true))
15    ));
16
17    //Checkbox button field
18    $fieldset-> add_field(new FormCheckbox('multiplechoice', array('title' => 'Answer2'),
19        new FormCheckboxOption(array('optiontitle' => 'Choix3', 'value' => 1)),
20        new FormCheckboxOption(array('optiontitle' => 'Choix4', 'value' => 2, 'checked'
=> true))
21    ));
22    //Select field
23    $fieldset-> add_field(new FormSelect('sex', array('title' => 'Sex'),
24        new FormSelectOption(array('optiontitle' => 'Men', 'value' => 1)),
25        new FormSelectOption(array('optiontitle' => 'Women', 'value' => 2)),
26        new FormSelectOption(array('optiontitle' => '?', 'value' => - 1, 'selected' =>
true))
27    ));
28
29    $form-> add_fieldset($fieldset); //Add fieldset to the form.
30
31    //##### Second fieldset #####//
32    $fieldset_up = new FormFieldset('Upload file');
33    //File field
34    $fieldset_up-> add_field(new FormFileUploader('avatar', array('title' => 'Avatar',
'subtitle' => 'Upload a file', 'class' => 'file', 'size' => 30)));
35    //Radio button field
36    $fieldset_up-> add_field(new FormHiddenField('test', array('value' => 1)));
37
38    $form-> add_fieldset($fieldset_up); //Add fieldset to the form.
39
40    $form-> display_preview_button('contents'); //Display a preview button for the textarea
field(ajax).
41    echo $form-> display(); //Display form.
```

- **Package** builder
- **Sub-Package** form
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

```
FIELD_INPUT__CHECKBOX = 'checkbox' [line 82]  
FIELD_INPUT__FILE = 'file' [line 84]  
FIELD_INPUT__HIDDEN = 'hidden' [line 83]  
FIELD_INPUT__RADIO = 'radio' [line 81]  
FIELD_INPUT__TEXT = 'text' [line 80]  
FIELD__SELECT = 'select' [line 86]  
FIELD__TEXTAREA = 'textarea' [line 85]
```

# Package builder Classes

## Class FormBuilder *[line 97]*

- **Package** builder
- **Sub-Package** form

### **FormBuilder::\$display\_preview**

*mixed = false [line 205]*

### **FormBuilder::\$display\_reset**

*mixed = true [line 208]*

### **FormBuilder::\$field\_identifier\_preview**

*mixed = 'contents' [line 206]*

### **FormBuilder::\$form\_action**

*mixed = " [line 203]*

### **FormBuilder::\$form\_class**

*mixed = 'fieldset\_mini' [line 204]*

### **FormBuilder::\$form\_fieldsets**

*mixed = array() [line 200]*

### **FormBuilder::\$form\_name**

*mixed = " [line 201]*

### **FormBuilder::\$form\_submit**

*mixed = " [line 202]*

Constructor *void* function FormBuilder::FormBuilder(\$form\_name, [\$form\_action = "], \$form\_title) *[line 105]*

**Function Parameters:**

- *string* **\$form\_name** The name of the form.
- *string* **\$form\_title** The title displayed for the form.
- *string* **\$form\_action** The url where the form send the data.

## constructor

*void* function FormBuilder::add\_fieldset(\$fieldset) *[line 118]*

**Function Parameters:**

- [FormFieldset](#) **\$fieldset** The fieldset object.

## Add fieldset in the form.

*string* function FormBuilder::display([\$Template = false]) *[line 128]*

**Function Parameters:**

- [Template](#) **\$Template** Optionnal template

## Return the form

*void* function FormBuilder::display\_preview\_button(\$field\_identifier\_preview, \$fieldset\_title) *[line 173]*

**Function Parameters:**

- *string* **\$fieldset\_title** The fieldset title
- **\$field\_identifier\_preview**

## Display a preview button for the specified field.

*void* function FormBuilder::display\_reset(\$value) *[line 183]*

**Function Parameters:**

- *boolean* **\$value** True to display, false to hide.

## Display a reset button for the form.

*void function FormBuilder::get\_form\_action() [line 197]*  
*void function FormBuilder::get\_form\_class() [line 198]*  
*void function FormBuilder::get\_form\_name() [line 195]*  
*void function FormBuilder::get\_form\_submit() [line 196]*  
*void function FormBuilder::set\_form\_action(\$form\_action) [line 191]*

### **Function Parameters:**

- **\$form\_action**

*void function FormBuilder::set\_form\_class(\$form\_class) [line 192]*

### **Function Parameters:**

- **\$form\_class**

*void function FormBuilder::set\_form\_name(\$form\_name) [line 189]*

### **Function Parameters:**

- **\$form\_name**

*void function FormBuilder::set\_form\_submit(\$form\_submit) [line 190]*

### **Function Parameters:**

- **\$form\_submit**

## Class FormCheckbox

*[line 36]*

**This class manages checkbox input fields.**

- **Package** builder
- **Sub-Package** form
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

### FormCheckbox::\$field\_options

*mixed = array() [line 91]*

Constructor *void* function FormCheckbox::FormCheckbox() *[line 38]*

*void* function FormCheckbox::add\_option(&\$option) *[line 60]*

#### **Function Parameters:**

- [FormRadioChoiceOption](#) &\$option option The new option.

### Add an option for the radio field.

*string* function FormCheckbox::display() *[line 68]*

## Class FormCheckboxOption

*[line 40]*

This class manages the checkbox fields. It provides you some additionnal field options:

- **optiontitle** : The option title
- **checked** : Specify it whether the option has to be checked.

- **Package** builder
- **Sub-Package** form
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

### FormCheckboxOption::\$option\_checked

*mixed = false [line 78]*

### FormCheckboxOption::\$option\_title

*mixed = " [line 77]*

Constructor *void* function FormCheckboxOption::FormCheckboxOption(\$field\_options) [line 42]

**Function Parameters:**

- **\$field\_options**

*string* function FormCheckboxOption::display() [line 66]

## Class FormField

[line 43]

**Abstract class which manage Fields. You can specify several option with the argument**

**\$fieldOptions :**

- **title :** The field title
- **subtitle :** The field subtitle
- **value :** The default value for the field
- **id :** The field identifier
- **class :** The css class used for the field
- **required :** Specify if the field is required.
- **onblur :** Action performed when cursor is clicked outside the field area. (javascript)

- **Package** builder
- **Sub-Package** form
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

### FormField::\$field\_css\_class

*mixed = " [line 137]*

### FormField::\$field\_errors

*mixed = array() [line 141]*

### FormField::\$field\_id

*mixed = " [line 136]*

### FormField::\$field\_name

*mixed = " [line 134]*

#### **FormField::\$field\_on\_blur**

*mixed = " [line 140]*

#### **FormField::\$field\_required**

*mixed = false [line 138]*

#### **FormField::\$field\_required\_alert**

*mixed = " [line 139]*

#### **FormField::\$field\_sub\_title**

*mixed = " [line 133]*

#### **FormField::\$field\_title**

*mixed = " [line 132]*

#### **FormField::\$field\_value**

*mixed = " [line 135]*

Constructor *void* function `FormField::FormField($fieldId, &$fieldOptions, $fieldOptions) [line 50]`

#### **Function Parameters:**

- *string* **\$fieldId** Name of the field.
- *array* **\$fieldOptions** Option for the field.
- **&\$fieldOptions**

### **constructor**

*void* function `FormField::add_errors(&$array_errors, $array_errors) [line 110]`

#### **Function Parameters:**

- *array* **\$array\_errors**
- **&\$array\_errors**

### **Merge errors.**

*array* function `FormField::get_errors() [line 119]`

**Get all errors occurred in the field construct process.**

*string* function FormField::get\_id() [line 124]  
*string* function FormField::get\_required\_alert() [line 129]  
*void* function FormField::throw\_error(\$errstr, \$errno) [line 101]

**Function Parameters:**

- *string* **\$errstr** Error message description
- *int* **\$errno** Error type, use php constants.

**Store all erros in the field construct process.**

## Class FormFieldset

[line 37]

- **Package** builder
- **Sub-Package** form
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>
- **Abstract Element**

### FormFieldset::\$fieldset\_display\_required

*mixed* = false [line 150]

### FormFieldset::\$fieldset\_errors

*mixed* = array() [line 149]

### FormFieldset::\$fieldset\_fields

*mixed* = array() [line 148]

### FormFieldset::\$fieldset\_title

*mixed* = " [line 147]

Constructor *void* function FormFieldset::FormFieldset(\$fieldset\_title, \$form\_name, \$form\_title, \$form\_action) [line 45]

**Function Parameters:**

- *string* **\$form\_name** The name of the form.
- *string* **\$form\_title** The tite displayed for the form.

- *string* **\$form\_action** The url where the form send the data.
- **\$fieldset\_title**

## constructor

*void* function FormFieldset::add\_field(\$form\_field) [*line 54*]

### **Function Parameters:**

- [FormField](#) **\$form\_field**

## Store fields in the fieldset.

*string* function FormFieldset::display([*\$Template* = false]) [*line 67*]

### **Function Parameters:**

- [Template](#) **\$Template** Optionnal template

## Return the form

*void* function FormFieldset::display\_preview\_button(\$field\_identifier\_preview, \$fieldset\_title) [*line 120*]

### **Function Parameters:**

- *string* **\$fieldset\_title** The fieldset title
- **\$field\_identifier\_preview**

## Display a preview button for the specified field.

*boolean* function FormFieldset::get\_display\_required() [*line 145*]

*array* function FormFieldset::get\_fields() [*line 140*]

*string* function FormFieldset::get\_title() [*line 135*]

*void* function FormFieldset::set\_display\_required(\$fieldset\_display\_required) [*line 130*]

### **Function Parameters:**

- *boolean* **\$fieldset\_display\_required**

*void* function FormFieldset::set\_title(\$fieldset\_title) [*line 125*]

### Function Parameters:

- *string* **\$fieldset\_title** The fieldset title

*void* function FormFieldset::throw\_error(\$errstr, \$errno) [*line 111*]

### Function Parameters:

- *string* **\$errstr** Error message description
- *int* **\$errno** Error type, use php constants.

**Store all erros in the field construct process.**

## Class FormFileUploader

[*line 39*]

**This class manage file input fields. It provides you additionnal field options :**

- **size** : The size for the field

- **Package** builder
- **Sub-Package** form
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

### FormFileUploader::\$field\_size

*mixed* = " [*line 85*]

Constructor *void* function FormFileUploader::FormFileUploader(\$fieldId, \$field\_options) [*line 41*]

### Function Parameters:

- **\$fieldId**
- **\$field\_options**

*string* function FormFileUploader::display() [*line 62*]

## Class FormHiddenField

[*line 35*]

This class manage hidden input fields.

- **Package** builder
- **Sub-Package** form
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

Constructor *void* function FormHiddenField::FormHiddenField(\$fieldId, \$field\_options) [*line 37*]

**Function Parameters:**

- **\$fieldId**
- **\$field\_options**

*void* function FormHiddenField::display() [*line 42*]

## Class FormRadioChoice

[*line 36*]

This class manage radio input fields.

- **Package** builder
- **Sub-Package** form
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

FormRadioChoice::\$field\_options

*mixed* = array() [*line 97*]

Constructor *void* function FormRadioChoice::FormRadioChoice(\$fieldId, \$fieldOptions, 2) [line 44]

**Function Parameters:**

- *string* **\$fieldId** Name of the field.
- *array* **\$fieldOptions** Option for the field.
- [FormRadioChoiceOption](#) **2** Variable number of FormRadioChoiceOption object to add in the FormRadioChoice.

**constructor** It takes a variable number of parameters. The first two are required.

*void* function FormRadioChoice::add\_option(&\$option) [line 66]

**Function Parameters:**

- [FormRadioChoiceOption](#) **&\$option** option The new option.

**Add an option for the radio field.**

*string* function FormRadioChoice::display() [line 74]

## Class FormRadioChoiceOption

[line 40]

**This class manage radio input field options. It provides you additional field options :**

- **optiontitle** : The option title
- **checked** : Specify if the option has to be checked.

- **Package** builder
- **Sub-Package** form
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

**FormRadioChoiceOption::\$option\_checked**

*mixed* = false [line 78]

## FormRadioChoiceOption::\$option\_title

*mixed = " [line 77]*

Constructor *void* function FormRadioChoiceOption::FormRadioChoiceOption(\$field\_options) *[line 42]*

### **Function Parameters:**

- **\$field\_options**

*string* function FormRadioChoiceOption::display() *[line 66]*

# Class FormSelect

*[line 40]*

**This class manage select fields. It provides you additionnal field options :**

- **multiple** : Type of select field, mutiple allow you to check several options.

- **Package** builder
- **Sub-Package** form
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

## FormSelect::\$field\_multiple

*mixed = false [line 112]*

## FormSelect::\$field\_options

*mixed = array() [line 111]*

Constructor *void* function FormSelect::FormSelect() *[line 42]*

*void* function FormSelect::add\_option(&\$option) *[line 74]*

### **Function Parameters:**

- [FormSelectOption](#) &\$option option The new option.

**Add an option for the radio field.**

*string* function FormSelect::display() [*line 82*]

## Class FormSelectOption

[*line 35*]

**This class manage select field options.**

- **Package** builder
- **Sub-Package** form
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

**FormSelectOption::\$option\_selected**

*mixed* = false [*line 72*]

**FormSelectOption::\$option\_title**

*mixed* = " [*line 71*]

Constructor *void* function FormSelectOption::FormSelectOption(\$field\_options) [*line 37*]

**Function Parameters:**

- **\$field\_options**

*string* function FormSelectOption::display() [*line 61*]

## Class FormTextarea

[*line 35*]

**This class manage multi-line text fields.**

- **Package** builder
- **Sub-Package** form

- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

#### FormTextarea::\$field\_cols

*mixed* = " [line 92]

#### FormTextarea::\$field\_editor

*mixed* = true [line 93]

#### FormTextarea::\$field\_forbidden\_tags

*mixed* = array() [line 94]

#### FormTextarea::\$field\_rows

*mixed* = " [line 91]

Constructor *void* function FormTextarea::FormTextarea(\$fieldId, \$fieldOptions) [line 37]

#### **Function Parameters:**

- **\$fieldId**
- **\$fieldOptions**

*string* function FormTextarea::display() [line 66]

## Class FormTextEdit

[line 41]

**This class manage single-line text fields. It provides you additional field options :**

- **size** : The maximum size for the field
- **maxlength** : The maximum length for the field
- **required\_alert** : Text displayed if field is empty (javascript only)

- **Package** builder
- **Sub-Package** form
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

### FormTextEdit::\$field\_maxlength

*mixed* = " [line 93]

### FormTextEdit::\$field\_size

*mixed* = " [line 92]

Constructor *void* function FormTextEdit::FormTextEdit(\$fieldId, \$field\_options) [line 43]

#### **Function Parameters:**

- **\$fieldId**
- **\$field\_options**

*string* function FormTextEdit::display() [line 67]



# Package content Procedural Elements

## categories\_manager.class.php

- **Package content**

```
ADD_THIS_CATEGORY_IN_LIST = true [line 41]
AJAX_MODE = false [line 32]
CATEGORY_DOES_NOT_EXIST = 0x08 [line 50]
CAT_UNVISIBLE = false [line 40]
CAT_VISIBLE = true [line 39]
DEBUG_MODE = true [line 29]
DISPLAYING_CONFIGURATION_NOT_SET = 0x20 [line 52]
DO_NOT_ADD_THIS_CATEGORY_IN_LIST = false [line 42]
DO_NOT_LOAD_CACHE = false [line 37]
ERROR_CAT_IS_AT_BOTTOM = 0x04 [line 49]
ERROR_CAT_IS_AT_TOP = 0x02 [line 48]
ERROR_UNKNOWN_MOTION = 0x01 [line 47]
IGNORE_AND_CONTINUE_BROWSING_IF_A_CATEGORY_DOES_NOT_MATCH = 2 [line 44]
INCORRECT_DISPLAYING_CONFIGURATION = 0x40 [line 53]
LOAD_CACHE = true [line 38]
MOVE_CATEGORY_DOWN = 'down' [line 36]
MOVE_CATEGORY_UP = 'up' [line 35]
NEW_CATEGORY_IS_IN_ITS_CHILDRENS = 0x80 [line 54]
NEW_PARENT_CATEGORY_DOES_NOT_EXIST = 0x10 [line 51]
NEW_STATUS_UNKNOWN = 0x100 [line 55]
NORMAL_MODE = true [line 31]
NOT_RECURSIVE_EXPLORATION = false [line 34]
PRODUCTION_MODE = false [line 30]
RECURSIVE_EXPLORATION = true [line 33]
STOP_BROWSING_IF_A_CATEGORY_DOES_NOT_MATCH = 1 [line 43]
```

# comments.class.php

- **Package** content

```
INTEGRATED_IN_ENVIRONMENT = true [line 28]  
KERNEL_SCRIPT = true [line 30]  
POP_UP_WINDOW = false [line 29]
```

# note.class.php

- **Package** content

```
NOTE_DISPLAY_BLOCK = 0x04 [line 30]  
NOTE_DISPLAY_NOTE = 0x01 [line 28]  
NOTE_NODISPLAY_NBRNOTES = 0x02 [line 29]
```

# search.class.php

- **Package** content

```
CACHE_TIME = $CONFIG['search_cache_time'] [line 31]  
CACHE_TIMES_USED = $CONFIG['search_max_use'] [line 32]
```

# Package content Classes

## Class CategoriesManager

[line 101]

This class enables you to manage easily the administration of categories for your modules. It's as generic as possible, if you want to complete some actions to specialize them for you module, you can create a new class inheriting of it in which you call its methods using the syntax `parent::method()`.

/\*! Warning : /\*!

- Your DB table must respect some rules :
  - You must have an integer attribute whose name is `id` and which represents the identifier of each category. It must be a primary key.
  - You also must have an integer attribute named `id_parent` which represents the identifier of the parent category (it will be 0 if its parent category is the root of the tree).
  - To maintain order, you must have a field containing the rank of the category which be an integer named `c_order`.
  - A field visible boolean (tynint 1 sur mysql)
  - A field name containing the category name
- In this class the user are supposed to be an administrator, no checking of his auth is done.
- To be correctly displayed, you must supply to functions a variable extracted from a file cache. Use the `Cache` class to build your file cache. Your variable must be an array in which keys are categories identifiers, an values are still arrays which are as this :
  - key `id_parent` containing the `id_parent` field of the database
  - key `name` containing the name of the category
  - key `order`
  - key `visible` which is a boolean
- You can also have other fields such as `auth level`, `description`, `visible`, that class won't modify them.
- To display the list of categories and actions you can do on them, you may want

to customize it. For that you must build an array that you will give to `set_display_config()` containing your choices :

- Key 'xmlhttprequest\_file' which corresponds to the name of the file which will treat the AJAX requests. We usually call it `xmlhttprequest.php`.
- Key 'url' which represents the url of the category (it won't display any link up to categories if you don't give this field). Its structure is the following :
  - key 'unrewritten' => string containing unrewritten urls (let %d where you want to display the category identifier)
  - Key `administration_file_name` which represents the file which allows you to update category
  - rewritten url (optional) 'rewritten' => string containing rewritten urls (let %d where you want to display the category identifier and %s the category name if you need it)

If you need more informations to use this class, we advise you to look at the wiki of PHPBoost, in which there is a tutorial explaining how to use it step by step.

- **Package** content
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

#### CategoriesManager::\$cache\_file\_name

```
string = " [line 853]
```

- **Var** name of the cache file of the module (usefull when this class regenerates it)

#### CategoriesManager::\$cache\_var

```
mixed[] = array() [line 868]
```

- **Var** Reference to the module cache variable containing the categories tree.

#### CategoriesManager::\$display\_config

*mixed[] = array()* [line 863]

- **Var** Current displaying configuration (see the class description to know its structure).

### CategoriesManager::\$errors

*int = 0* [line 858]

- **Var** Current error status

### CategoriesManager::\$table

*string = "* [line 848]

- **Var** table name where are stocked the categories to manage (see the class description for more details).

Constructor *void* function CategoriesManager::CategoriesManager(\$table, \$cache\_file\_name, &\$cache\_var) [line 110]

#### **Function Parameters:**

- *string* **\$table** Table name of the database which contains the require fields (explained in the class description)
- *string* **\$cache\_file\_name** Name of the cache file (usefull to regenerate the cache after a modification of the categories tree)
- *&array[]* **&\$cache\_var** Array containing the correct data, described in the description of the class.

### Builds a CategoriesManager object

*int* function CategoriesManager::add(\$id\_parent, \$name, [\$visible = CAT\_VISIBLE], [\$order = 0]) [line 127]

#### **Function Parameters:**

- *int* **\$id\_parent** Id of the category in which this category will be added
- *string* **\$name** Name of the category to add
- *bool* **\$visible** Is the category visible? CAT\_VISIBLE if visible, CAT\_UNVISIBLE else
- *int* **\$order**

## Adds a category. We can decide if it will be visible and what its position will be

*mixed* function CategoriesManager::build\_administration\_interface([\$ajax\_mode = NORMAL\_MODE], [\$category\_template = NULL]) [*line 437*]

### Function Parameters:

- *bool* **\$ajax\_mode** Set this parameter to NORMAL\_MODE if it's the global display and AJAX\_MODE if it's called in the AJAX handler.
- *Template* **\$category\_template** Use this parameter if you want to use a particular template. The default theme is framework/content/category.tpl.

**Builds the list of categories and links to makes operations to administrate them (delete, move, add...), it supplies a string ready to be displayed. It uses AJAX, read the class description to understand this user interface.**

- **Warning** You must have defined a displaying configuration before calling this method. You must do it with the set\_display\_config method.

*int[]* function CategoriesManager::build\_children\_id\_list(\$category\_id, &\$list, [\$recursive\_exploration = RECURSIVE\_EXPLORATION], [\$add\_this = DO\_NOT\_ADD\_THIS\_CATEGORY\_IN\_LIST], [\$num\_auth = 0]) [*line 521*]

### Function Parameters:

- *int* **\$category\_id** Id of the category for which we want to know the children
- *&int[]* **&\$list** Array in which the result will be written.
- *bool* **\$recursive\_exploration** Sets if you want to explorer only the current sublevel of the tree or the whole subtree. Use RECURSIVE\_EXPLORATION to make a recursive exploration, NOT\_RECURSIVE\_EXPLORATION otherwise.
- *bool* **\$add\_this** If you want to add the current category to the list use ADD\_THIS\_CATEGORY\_IN\_LIST, DO\_NOT\_ADD\_THIS\_CATEGORY\_IN\_LIST otherwise.
- *int* **\$num\_auth** If you want to filter the category according to an authorization bit, put its value here

## Builds the list of all the children of a category

*int[]* function CategoriesManager::build\_parents\_id\_list(\$category\_id, [\$add\_this = DO\_NOT\_ADD\_THIS\_CATEGORY\_IN\_LIST]) [*line 558*]

### Function Parameters:

- *int* **\$category\_id** Id of the category of which you want the list of parents categories.
- *bool* **\$add\_this** If you want to add the current cat at the list. Use ADD\_THIS\_CATEGORY\_IN\_LIST if you want, DO\_NOT\_ADD\_THIS\_CATEGORY\_IN\_LIST otherwise.

## Builds the list of the parent categories of a category

*string* function CategoriesManager::build\_select\_form(\$selected\_id, \$form\_id, \$form\_name, [\$current\_id\_cat = 0], [\$num\_auth = 0], [\$array\_auth = array()], [\$recursion\_mode = STOP\_BROWSING\_IF\_A\_CATEGORY\_DOES\_NOT\_MATCH], [\$template = NULL]) [*line 488*]

### Function Parameters:

- *int* **\$selected\_id** Current category id (the id of the category selected in default displaying).
- *string* **\$form\_id** HTML identifier of the object (id of the DOM Document Objet Model).t'
- *string* **\$form\_name** HTML name of the select form in which you will manage to retrieve the selected category.
- *int* **\$current\_id\_cat** This parameter is to use when for instance you want to move a category in another, it will not display the children categories of the current category because you cannot move it into one of its subcategories.
- *int* **\$num\_auth** If you don't want to display the categories which can not be chosen by the user, you can supply an authorization number which will be used on the \$array\_auth parameter
- *mixed[]* **\$array\_auth** Authorization array which is used if a category hasn't special authorizations
- *bool* **\$recursion\_mode** Sets whether you want to display only the current category or its whole subcategories. Use the constant RECURSIVE\_EXPLORATION if you want to explore the whole tree, NOT\_RECURSIVE\_EXPLORATION otherwise.
- *Template* **\$template** If you want to customize the displaying, you can give the method a template objet in which variables will be assigned. The default template is framework/content/categories\_select\_form.tpl.

## Builds a formulary which allows user to choose a category in a select form.

*bool* function CategoriesManager::change\_visibility(\$category\_id, \$visibility, [\$generate\_cache = LOAD\_CACHE]) [*line 345*]

### Function Parameters:

- *int* **\$category\_id** id of the category whose property must be changed
- *bool* **\$visibility** set to visible or unvisible (use constants CAT\_VISIBLE and CAT\_UNVISIBLE)
- *bool* **\$generate\_cache** if you want that the system regenerate the cache file of the module. Use

the constants `LOAD_CACHE` to regenerate and reload the cache or `DO_NOT_LOAD_CACHE` else.

## Changes the visibility of a category

*bool* function `CategoriesManager::check_display_config([$debug = PRODUCTION_MODE])` *[line 400]*

### **Function Parameters:**

- *bool* **\$debug** `DEBUG_MODE` if you want to display the errors, or `PRODUCTION_MODE` to return false if it fails.

## Checks if display configuration is good

*bool* function `CategoriesManager::check_error($error)` *[line 580]*

### **Function Parameters:**

- *int* **\$error** Constant corresponding to the error to check. Use the constant corresponding to the error (detailed in each method description).

**Checks if an error has been raised on the last reported error. At each call of a method of this class which can raise an error, the last error is erased.**

*mixed[]* function `CategoriesManager::compute_heritated_auth($category_id, $bit, $mode)` *[line 592]*

### **Function Parameters:**

- *int* **\$category\_id** Id of the category for which you want to know what is the global authorization
- *int* **\$bit** The authorization bit you want to check
- *int* **\$mode** Merge mode. If it corresponds to a read authorization, use `AUTH_PARENT_PRIORITY` which will disallow for example all the subcategories of a category to which you can't access, or `AUTH_CHILD_PRIORITY` if you want to work in write mode, each child will be able to redefine the authorization.

**Computes the global authorization level of the whole parent categories. The result corresponds to all the category's parents merged.**

*bool* function `CategoriesManager::delete($id)` *[line 308]*

### **Function Parameters:**

- *int* **\$id** Id of the category to delete.

### Deletes a category.

*FeedsList* function `CategoriesManager::get_feeds_list()` [*line 614*]

**Computes the list of the feeds corresponding to each category of the category tree.**

*bool* function `CategoriesManager::move($id, $way)` [*line 171*]

#### **Function Parameters:**

- *int* **\$id** Id of the category to move
- *string* **\$way** The way according to which the category has to be moved. It must be either MOVE\_CATEGORY\_UP or MOVE\_CATEGORY\_DOWN.

### Moves a category (makes it gone up or down)

*bool* function `CategoriesManager::move_into_another($id, $new_id_cat, [$position = 0])` [*line 248*]

#### **Function Parameters:**

- *int* **\$id** Id of the category to move
- *int* **\$new\_id\_cat** Id of the parent category in which the category will be moved.
- *int* **\$position** Position (number) that the category has to take in its new parent category. If not specified, it will be placed at the end of the category.

### Moves a category into another category. You can specify its future position in its future parent category.

*unknown\_type* function `CategoriesManager::set_display_config($config)` [*line 387*]

#### **Function Parameters:**

- *\$config* **\$config**

**Method which sets the displaying configuration** Config example `$config = array(`  
`'xmlhttprequest_file' => 'xmlhttprequest.php',                    'administration_file_name'`  
`=> 'admin_news_cats.php',                    'url' => array(                    'unrewrited' =>`  
`PATH_TO_ROOT . '/news/news.php?id=%d',                    'rewrited' =>`  
`PATH_TO_ROOT . '/news-%d+%s.php'                    ), );`

# Class Comments

[line 40]

This class manages comments everywhere in phpboost Simplified use with the `display_comments` function: //news is the name of the modue, \$idnews is the id in database for this item. `display_comments('news', $idnews, url('news.php?id=' . $idnews . '&com=%s', 'news-0-' . $idnews . '.php?com=%s'))`

- **Package** content
- **Author** Révis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

## Comments::\$idcom

*mixed = 0 [line 739]*

## Comments::\$idprov

*mixed = 0 [line 738]*

## Comments::\$is\_kernel\_script

*mixed = false [line 746]*

## Comments::\$lock\_com

*mixed = 0 [line 745]*

## Comments::\$module\_folder

*mixed = " [line 742]*

## Comments::\$nbr\_com

*mixed = 0 [line 744]*

## Comments::\$path

*mixed = " [line 740]*

## Comments::\$script

*mixed = " [line 737]*

## Comments::\$sql\_table

*mixed* = " [line 743]

## Comments::\$vars

*mixed* = " [line 741]

Constructor *void* function Comments::Comments(\$script, \$idprov, \$vars, [\$module\_folder = ""], [\$is\_kernel\_script = false]) [line 51]

### Function Parameters:

- *string* **\$script** Module's name
- *string* **\$idprov** Id field in the database. related to item where the comments are posted.
- *string* **\$vars** Link for the module, it has to include com=%s in the link for the comments system.
- *string* **\$module\_folder** Module's folder
- *string* **\$is\_kernel\_script**

## Display comments form.

*int* function Comments::add(\$contents, \$login) [line 65]

### Function Parameters:

- *string* **\$contents** Comment content
- *string* **\$login** Poster's login

## Add a comment

*string* function Comments::com\_display\_link(\$nbr\_com, \$path, \$idprov, \$script, [\$options = 0]) [line 688]

### Function Parameters:

- *int* **\$nbr\_com**
- *string* **\$path**
- *int* **\$idprov**
- *string* **\$script**
- *int* **\$options**

- **Static**

*int* function Comments::del() [*line 93*]

### Delete a comment

*void* function Comments::delete\_all(\$idprov) [*line 118*]

#### Function Parameters:

- *int* **\$idprov** The id field of the item in the database.

### Delete all comments for the specified item.

*string* function Comments::display([\$integrated\_in\_environment = INTEGRATED\_IN\_ENVIRONMENT],  
[\$Template = false], [\$page\_path\_to\_root = ""]) [*line 182*]

#### Function Parameters:

- *int* **\$integrated\_in\_environment**
- [\*Template\*](#) **\$Template** Optional template.
- *string* **\$page\_path\_to\_root**

### Display comments form.

*void* function Comments::get\_attribute(\$varname) [*line 170*]

#### Function Parameters:

- **\$varname**

*boolean* function Comments::is\_loaded() [*line 140*]

### Check if the comments system is correctly loaded.

*void* function Comments::lock(\$lock) [*line 129*]

#### Function Parameters:

- *boolean* **\$lock** true for locking, false otherwise

### Lock or unlock comments for an item.

*void function Comments::set\_arg(\$idcom, [\$path = "]) [line 155]*

**Function Parameters:**

- *int* **\$idcom**
- *string* **\$path**

**Set argument for the comments system.**

*void function Comments::update(\$contents, \$login) [line 83]*

**Function Parameters:**

- *string* **\$contents** Comment content
- *string* **\$login** Poster's login

**Edit a comment**

## Class Note

*[line 37]*

**This class provides you an easy way to manage notation.**

- **Package** content
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

**Note::\$idprov**

*mixed = 0 [line 277]*

**Note::\$module\_folder**

*mixed = " [line 280]*

**Note::\$notation\_scale**

*mixed = " [line 283]*

### Note::\$options

*mixed* = " [line 281]

### Note::\$path

*mixed* = " [line 278]

### Note::\$script

*mixed* = " [line 276]

### Note::\$script\_path

*mixed* = " [line 279]

### Note::\$sql\_table

*mixed* = " [line 282]

Constructor *void* function Note::Note(\$script, \$idprov, \$script\_path, \$notation\_scale, [\$module\_folder = "], [\$options = 0]) [line 50]

#### Function Parameters:

- *string* **\$script** The module name in which the pagination is used.  
<strong>Warning</strong>The sql table name has to be the same name, otherwise you has to specify the sql table name in \$script and module name in \$module\_folder argument
- *int* **\$idprov** The id of the item in the module.
- *string* **\$script\_path** The script path. Example : url('file.php?id=' . \$idart, 'file-' . \$idart . '.php')
- *int* **\$notation\_scale** The notation scale
- *string* **\$module\_folder** (optional) The folder where the module is located. It allow you to specify a different module location
- *int* **\$options**

## Create an new object Note.

Return function Note::add(\$note) [line 63]

#### Function Parameters:

- *int* **\$note** The note given by the user.

## Add a note on the item. It has to be into the notation scale.

*string* function Note::display\_form([\$template = false]) [line 94]

#### Function Parameters:

- *object* **\$template** Template object to use another template file.

## Display the ajax notation form.

*string* function Note::display\_img(\$note, \$notation\_scale, [\$num\_stars\_display = 0]) [line 192]

### **Function Parameters:**

- *int* **\$note** The note of the item.
- *int* **\$notation\_scale** Notation scale
- *int* **\$num\_stars\_display** Number of image to display.

**Static method which display the notation with images, you can restrain the number of images displayed with the argument \$num\_stars\_display**

- **Static**

*unknown\_type* function Note::get\_attribute(\$varname) [line 231]

### **Function Parameters:**

- *string* **\$varname**

## Accessor

## Class Search

[line 39]

- **Package** content
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >

### Search::\$errors

*mixed* = [line 460]

### Search::\$id\_search

*mixed* = [line 455]

### Search::\$id\_user

*mixed* = [line 459]

### Search::\$modules

*mixed* = [line 457]

### Search::\$modules\_conditions

*mixed* = [line 458]

### Search::\$search

*mixed* = [line 456]

Constructor *void* function Search::Search([\$search = "], [\$modules = array()]) [line 53]

#### **Function Parameters:**

- *string* **\$search** the string to search
- *mixed[]* **\$modules** Modules in which we gonna search with their search params. This argument is an array which keys are module id's and values are arrays containing the specialized search arguments for a particular module.

**Builds a search object. Query Complexity: 6 + k / 10 database queries. (k represent the number of module without search cache)**

*int* function Search::get\_ids() [line 411]

#### **Returns the search id**

*int* function Search::get\_results(&\$results, &\$module\_ids, [\$nb\_lines = 0], [\$offset = 0]) [line 229]

#### **Function Parameters:**

- *string[]* **&\$results** the results returned
- *string[]* **&\$module\_ids** the modules ids from which retrieve results
- *int* **\$nb\_lines** the number of lines to return
- *int* **\$offset** the offset from which return results

**Puts results from the search results in the \$results parameter and returns the number of results. Query complexity: 1 query.**

*int* function Search::get\_results\_by\_id(&\$results, [\$id\_search = 0], [\$nb\_lines = 0], [\$offset = 0]) [*line 191*]

**Function Parameters:**

- *string[]* **&\$results** the results returned
- *int* **\$id\_search** the search id
- *int* **\$nb\_lines** the number of lines to return
- *int* **\$offset** the offset from which return results

**Puts results from the search results identified by the \$id\_search parameter in the \$results parameter and returns the number of results. Query complexity: 2 queries.**

*void* function Search::insert\_results(&\$requestAndResults) [*line 287*]

**Function Parameters:**

- *mixed[]* **&\$requestAndResults** This parameters is an array with keys that are modules ids and values that could be both a SQL query or a results array.

**Inserts search results in the database cache in order to speed up next searches. Query complexity: 1 + k / 10 queries. (k represent the number of results to insert in the database)**

*bool* function Search::is\_in\_cache(\$module\_id) [*line 392*]

**Function Parameters:**

- *string* **\$module\_id** the module to check.

**Returns true if the module results are in cache, else, false.**

*bool* function Search::is\_search\_id\_in\_cache(\$id\_search) [*line 365*]

**Function Parameters:**

- *int* **\$id\_search** the search id to check.

**Returns true if the id\_search is in cache, else, false.**

*string[]* function Search::modules\_in\_cache() [*line 402*]

**Returns the list of the modules ids present in the cache**

## Class BBCodeEditor

[*line 36*]

**This class provides an interface editor for contents.**

- **Package** content
- **Sub-Package** editor
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

Constructor *void* function BBCodeEditor::BBCodeEditor() [*line 38*]

*string* function BBCodeEditor::display() [*line 47*]

**Display the editor**

## Class ContentEditor

[*line 34*]

**Abstract class for editors content.**

- **Package** content
- **Sub-Package** editor
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

**ContentEditor::\$forbidden\_tags**

*mixed* = array() [line 101]

#### **ContentEditor::\$identifier**

*mixed* = 'contents' [line 102]

#### **ContentEditor::\$language\_type**

*mixed* = DEFAULT\_LANGUAGE [line 100]

#### **ContentEditor::\$template**

*mixed* = null [line 103]

Constructor *void* function ContentEditor::ContentEditor([\$language\_type = null]) [line 36]

#### **Function Parameters:**

- **\$language\_type**

*array* function ContentEditor::get\_forbidden\_tags() [line 60]

#### **Get the forbidden tags.**

*Template* function ContentEditor::get\_template() [line 87]

#### **Get the template used for the editor.**

*void* function ContentEditor::set\_forbidden\_tags(&\$forbidden\_tags) [line 51]

#### **Function Parameters:**

- *array* **&\$forbidden\_tags** List of forbidden tags.

#### **Set the forbidden tags**

*void* function ContentEditor::set\_identifier(\$identifier) [line 69]

#### **Function Parameters:**

- *string* **\$identifier** The html identifier.

#### **Set the html identifier of the textarea field which contain the content to edit.**

*void* function ContentEditor::set\_template(&\$template, \$template.) [line 78]

#### **Function Parameters:**

- [Template](#) \$template.
- `&$template`

Set an alternative template for the editor.

## Class TinyMCEEditor

[line 36]

This class provides an interface editor for contents.

- **Package** content
- **Sub-Package** editor
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

### TinyMCEEditor::\$array\_tags

```
mixed = array('align1' => 'justifyleft', 'align2' => 'justifycenter', 'align3' => 'justifyright', 'align4' => 'justifyfull',
'|1' => '|', 'title' => 'formatselect', '2' => '|', 'list1' => 'bullist', 'list2' => 'numlist', '3' => '|', 'indent1' => 'outdent',
'indent2' => 'indent', '4' => '|', 'quote' => 'blockquote', 'line' => 'hr', '5' => '|', '_cut' => 'cut', '_copy' => 'copy',
'_paste' => 'paste', '6' => '|', '_undo' => 'undo', '_redo' => 'redo') [line 106]
```

### TinyMCEEditor::\$array\_tags2

```
mixed = array('b' => 'bold', 'i' => 'italic', 'u' => 'underline', 's' => 'strikethrough', '1' => '|', 'color1' =>
'forecolor', 'color2' => 'backcolor', '1' => '|', '2' => '|', 'size' => 'fontsizeselect', 'font' => 'fontselect', '3' => '|',
'sub' => 'sub', 'sup' => 'sup', '4' => '|', 'url1' => 'link', 'url2' => 'unlink', '5' => '|', 'img' => 'image', 'swf' =>
'flash') [line 107]
```

### TinyMCEEditor::\$array\_tags3

```
mixed = array('emotions' => 'emotions', 'table' => 'tablecontrols', '2' => '|', 'image', 'anchor' => 'anchor',
'_charmap' => 'charmap', '3' => '|', '_cleanup' => 'cleanup', '_removeformat' => 'removeformat', '4' => '|',
'_search' => 'search', '_replace' => 'replace', '5' => '|', '_fullscreen' => 'fullscreen') [line 108]
```

Constructor *void* function TinyMCEEditor::TinyMCEEditor() [line 38]

*string* function TinyMCEEditor::display() [line 47]

### Display the editor

# bbcode\_highlighter.class.php

- **Package** content
- **Sub-Package** parser

```
BBCODE_HIGHLIGHTER_BLOCK_CODE = false [line 42]
BBCODE_HIGHLIGHTER_INLINE_CODE = true [line 41]
BBCODE_LIST_ITEM_COLOR = '#00AF07' [line 38]
BBCODE_PARAM_COLOR = '#7B00FF' [line 34]
BBCODE_PARAM_NAME_COLOR = '#FF0000' [line 36]
BBCODE_TAG_COLOR = '#0000FF' [line 32]
```

## content\_formatting\_factory.class.php

- **Package** content
- **Sub-Package** parser

```
BBCODE_LANGUAGE = 'bbcode' [line 28]  
DEFAULT_LANGUAGE = 'default' [line 30]  
TINYMCE_LANGUAGE = 'tinymce' [line 29]
```

# parser.class.php

- **Package** content
- **Sub-Package** parser

```
ADD_SLASHES = true [line 30]  
DO_NOT_ADD_SLASHES = false [line 29]  
PARSER_DO_NOT_STRIP_SLASHES = false [line 31]  
PARSER_STRIP_SLASHES = true [line 32]  
PICK_UP = true [line 33]  
REIMPLANT = false [line 34]
```

# template\_highlighter.class.php

- **Package** content
- **Sub-Package** parser

```
TPL_BRACES_STYLE = 'color:#7F3300;' [line 31]
TPL_KEYWORD_STYLE = 'color:#000066; font-weight: bold;' [line 35]
TPL_NESTED_VARIABLE_STYLE = 'color:#8F5211;' [line 33]
TPL_SHARP_STYLE = 'color:#9915AF; font-weight: bold;' [line 34]
TPL_VARIABLE_STYLE = 'color:#FF6600; font-weight: bold;' [line 32]
```

## Class BBCodeHighlighter

[line 51]

This class is a highlighter for the PHPBoost BBCode language. It supplies the highlighted code written in XHTML.

- **Package** content
- **Sub-Package** parser
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

Constructor *void* function BBCodeHighlighter::BBCodeHighlighter() [line 57]

**Builds a BBCodeHighlighter objet**

*void* function BBCodeHighlighter::parse([\$inline\_code = BBCodeHighlighter::BLOCK\_CODE]) [line 69]

**Function Parameters:**

- *bool* **\$inline\_code** If you want that the code make a new HTML paragraph, use BBCodeHighlighter::BLOCK\_CODE (default parameter) and if you want that it would be integrated in a line, use BBCodeHighlighter::INLINE\_CODE

**Highlights the content of the parser.**

# Class BBCodeParser

[line 38]

Converts the PHPBoost BBCode language to the XHTML language which is stocked in the database and can be displayed nearly directly. It parses only the authorized tags (defined in the parent class which is ContentParser).

- **Package** content
- **Sub-Package** parser
- **Author** Benoit Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com)>

Constructor *void* function BBCodeParser::BBCodeParser() [line 43]

**Builds a BBCodeParser object**

*string* function BBCodeParser::clear\_html\_br(\$matches) [line 484]

**Function Parameters:**

- *string[]* **\$matches** Content matched by a regular expression

**Callback which clears the new line tag in the HTML generated code**

*void* function BBCodeParser::parse() [line 52]

**Parses the parser content from BBCode to XHTML.**

# Class BBCodeUnparser

[line 37]

BBCode unparser. It converts a content using the PHPBoost HTML reference code (for example coming from a database) to the PHPBoost BBCode syntax.

- **Package** content
- **Sub-Package** parser
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com)>

Constructor *void* function BBCodeUnparser::BBCodeUnparser() [*line 42*]

**Builds a BBCodeUnparser object**

*void* function BBCodeUnparser::parse() [*line 52*]

**Unparses the content of the parser. Converts it from HTML syntax to BBcode syntax**

## Class ContentFormattingFactory

[*line 40*]

This class is approximatively a factory which provides objets capable to format some content. The text formatting uses a syntax, PHPBoost supports both the BBCode syntax and a WYSIWYG tool syntax (TinyMCE). You can choose the formatting type you want to deal with.

- **Package** content
- **Sub-Package** parser
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com)>

**ContentFormattingFactory::\$language\_type**

*string* = DEFAULT\_LANGUAGE [*line 250*]

- **Var** Language type

Constructor *void* function ContentFormattingFactory::ContentFormattingFactory([\$language\_type = false]) [*line 52*]

### Function Parameters:

- **string \$language\_type** The language in which must work the factory. One of the following elements:
  - `BBCODE_LANGUAGE` if you want to force the BBCode language
  - `TINYMCE_LANGUAGE` if you want to force the TinyMCE formatting editor
  - `DEFAULT_LANGUAGE` if you want to use the default language of the site. If the user changed it, it will be the language he chose, otherwise it will be the website default language.

## Builds a ContentFormattingFactory object

*string[]* function ContentFormattingFactory::get\_available\_tags() [*line 204*]

**Returns the map of all the formatting types supported by the PHPBoost formatting editors and parsers. The keys of the map are the tags identifiers and the values the tags names.**

- **Static**

*ContentParser* function ContentFormattingFactory::get\_editor() [*line 164*]

**Returns an editor object which will display the editor corresponding to the language you chose.**

*string* function ContentFormattingFactory::get\_language() [*line 87*]

**Returns the language of the factory**

*ContentParser* function ContentFormattingFactory::get\_parser() [*line 96*]

**Returns a parser which will work in the language you chose.**

*ContentSecondParser* function ContentFormattingFactory::get\_second\_parser() [*line 154*]

**Returns a second parser which will work in the language you chose.**

*ContentUnparser* function ContentFormattingFactory::get\_unparser() [*line 125*]

**Returns a unparser which will work in the language you chose.**

*string* function ContentFormattingFactory::get\_user\_editor() [*line 192*]

Returns the name of the editor of the current user (chosen in its profile).

*void* function ContentFormattingFactory::set\_language([\$language\_type = DEFAULT\_LANGUAGE]) [*line 70*]

**Function Parameters:**

- **string \$language\_type** The language in which must work the factory. One of the following elements:
  - BBCode\_LANGUAGE if you want to force the BBCode language
  - TINYMCE\_LANGUAGE if you want to force the TinyMCE formatting editor
  - DEFAULT\_LANGUAGE if you want to use the default language of the site. If the user changed it, it will be the language he chosed, otherwise it will be the website default language.

Changes the language of the factory

## Class ContentParser

[*line 36*]

This class is abstract. It contains tools that are usefull for implement a content parser.

- **Package** content
- **Sub-Package** parser
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

**ContentParser::\$forbidden\_tags**

*string[]* = array() [*line 319*]

- **Var** List of the BBCode forbidden tags

**ContentParser::\$html\_auth**

*string[]* = array() [*line 315*]

- **Var** Authorization of the HTML BBCode tag.

### **ContentParser::\$tag**

```
mixed = array('b', 'i', 'u', 's', 'title', 'stitle', 'style', 'url',  
             'img', 'quote', 'hide', 'list', 'color', 'bgcolor', 'font', 'size', 'align', 'float', 'sup',  
             'sub', 'indent', 'pre', 'table', 'swf', 'movie', 'sound', 'code', 'math', 'anchor', 'acronym') [line 309]
```

Constructor *void* function ContentParser::ContentParser() [*line 42*]

**Buils a ContentParser object.**

*string[]* function ContentParser::get\_forbidden\_tags() [*line 77*]

**Gets the forbidden tags.**

*void* function ContentParser::parse() [*line 55*]

**Parses the content of the parser**

*void* function ContentParser::set\_forbidden\_tags(\$forbidden\_tags) [*line 65*]

**Function Parameters:**

- *string[]* **\$forbidden\_tags** list of the name of the tags which mustn't be parsed.

**Sets the tags which mustn't be parsed.**

*void* function ContentParser::set\_html\_auth(\$array\_auth) [*line 87*]

**Function Parameters:**

- *mixed[]* **\$array\_auth** authorization array

**Sets the required authorizations that are necessary to post some HTML code which will be displayed by the web browser.**

# Class ContentSecondParser

*[line 40]*

This class ensures the real time processing of the content. The major part of the processing is saved in the database to minimize as much as possible the treatment when the content is displayed. However, some tags cannot be cached, because we cannot have return to the original code. It's for instance the case of the code tag which replaces the code by a lot of html code which formats the code. This kind of tag is treated in real time by this class. The content you put in that parser must come from a ContentParser class (BBCodeParser or TinyMCEParser) (it can have been saved in a database between the first parsing and the real time parsing).

- **Package** content
- **Sub-Package** parser
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

Constructor *void* function ContentSecondParser::ContentSecondParser() *[line 46]*

**Builds a ContentSecondParser object**

*string* function ContentSecondParser::export\_html\_text(\$html\_content, \$html) *[line 86]*

**Function Parameters:**

- *string* **\$html** Content to transform
- **\$html\_content**

**Transforms a PHPBoost HTML content to make it exportable and usable every where in the web.**

*void* function ContentSecondParser::parse() *[line 54]*

**Parses the content of the parser. The result will be ready to be displayed.**

# Class ContentUnparser

[line 36]

This class is an abstract class. It contains the common elements needed by all the unparsers of PHPBoost.

- **Package** content
- **Sub-Package** parser
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

Constructor *void* function ContentUnparser::ContentUnparser() [line 42]

**Builds a ContentUnparser class.**

# Class Parser

[line 42]

This class is the basis of all the formatting processings that exist in PHPBoost.

- **Package** content
- **Sub-Package** parser
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

**Parser::\$array\_tags**

*string[] = array()* [line 135]

- **Var** List of the tags which have been picked up by the parser
- **Static**

**Parser::\$content**

*string = " [line 130]*

- **Var** Content of the parser

#### **Parser::\$page\_path**

*string = " [line 158]*

- **Var** Path of the page in which has been written the content to parse.

#### **Parser::\$path\_to\_root**

*string = PATH\_TO\_ROOT [line 153]*

- **Var** Path to root of the page in which has been written the content to parse.

Constructor *void* function `Parser::Parser()` *[line 48]*

**Builds a Parser object.**

*string* function `Parser::get_content([$addslashes = ADD_SLASHES])` *[line 60]*

**Function Parameters:**

- *bool* **\$addslashes** `ADD_SLASHES` if you want to escape the slashes in your string (you often save a parsed content into the database when you parse it), otherwise `DO_NOT_ADD_SLASHES`.

**Returns the content of the parser. If you called a method which parses the content, this content will be parsed.**

*string* function `Parser::get_page_path()` *[line 121]*

**Returns the page path**

Returns the page path

*string* function Parser::get\_path\_to\_root() [*line 103*]

### Returns the path to root attribute.

Returns the path to root attribute.

*void* function Parser::set\_content(\$content, [\$stripslashes = PARSEER\_DO\_NOT\_STRIP\_SLASHES]) [*line 78*]

#### Function Parameters:

- *string* **\$content** Content
- *bool* **\$stripslashes** PARSEER\_DO\_NOT\_STRIP\_SLASHES if you don't want to strip slashes before adding it to the parser, otherwise PARSEER\_STRIP\_SLASHES.

**Sets the content of the parser. When you will call a parse method, it will deal with this content.**

*void* function Parser::set\_page\_path(\$page\_path) [*line 112*]

#### Function Parameters:

- *string* **\$page\_path** Page path

### Sets the page path

Sets the page path

*void* function Parser::set\_path\_to\_root(\$path) [*line 94*]

#### Function Parameters:

- *string* **\$path** Path

### Sets the reference path for relative URL

Sets the reference path for relative URL

# Class TemplateHighlighter

[line 43]

This is a syntax highlighter for the PHPBoost template syntax.

- **Package** content
- **Sub-Package** parser
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com)>

Constructor *void* function TemplateHighlighter::TemplateHighlighter() [line 49]

**Build a TemplateHighlighter object.**

*void* function TemplateHighlighter::parse([\$line\_number = GESHI\_NO\_LINE\_NUMBERS], [\$inline\_code = false]) [line 59]

**Function Parameters:**

- *int* **\$line\_number** GESHI\_NO\_LINE\_NUMBERS => no line numbers, GESHI\_NORMAL\_LINE\_NUMBERS line numbers.
- *bool* **\$inline\_code** true if it's a single line code, otherwise false.

**Highlights the code. It uses the geshi HTML syntax highlighter and then it highlights the specific template syntax.**

# Class TinyMCEParser

[line 40]

This class enables to use TinyMCE without breaking the compatibility with the BBCode formatting. PHPBoost has a reference syntax, it in HTML with specific CSS classes. The HTML code generated by TinyMCE must be modified to conform itself to this specific syntax. This class makes the translation from the TinyMCE HTML to the PHPBoost HTML.

- **Package** content
- **Sub-Package** parser
- **Author** Benoit Sautel
- **See** The TinyMCEUnparser class which makes the reverse operation.

Constructor *void* function TinyMCEParser::TinyMCEParser() [*line 45*]

**Builds this kind of parser**

*void* function TinyMCEParser::parse() [*line 54*]

**Parses the content of the parser. Translates the whole content from the TinyMCE syntax to the PHPBoost one.**

## Class TinyMCEUnparser

[*line 41*]

This class enables to translate the content formatting from the PHPBoost standard one to the TinyMCE one. The PHPBoost one is historically the one corresponding to the BBCode translation in HTML and is now the reference. TinyMCE has a particular syntax and it must be respected if we want to make a formatting which can be edited after having been written, enough what using a WYSIWYG editor hasn't any advantage.

- **Package** content
- **Sub-Package** parser
- **Author** Benoit Sautel ben.popeye@phpboost.com
- **See** the TinyMCEParser class which makes the reverse operation.

Constructor *void* function TinyMCEUnparser::TinyMCEUnparser() [*line 46*]

**Builds a TinyMCEUnparser object**

*void* function TinyMCEUnparser::parse() [*line 55*]

**Unparses the content of the parser. It goes from the PHPBoost reference formatting syntax to the TinyMCE one.**

# site\_map.class.php

- **Package** content
- **Sub-Package** sitemap

`SITE_MAP_AUTH_GUEST = false` *[line 39]*

**The site map will be seen by every body, only the public elements must appear**

The site map will be seen by every body, only the public elements must appear

`SITE_MAP_AUTH_USER = true` *[line 43]*

**The site map is for the current user.**

The site map is for the current user. It must contain only what the user can see, but it can be private.

`SITE_MAP_FREQ_ALWAYS = 'always'` *[line 57]*

`SITE_MAP_FREQ_DAILY = 'daily'` *[line 59]*

`SITE_MAP_FREQ_DEFAULT = SITE_MAP_FREQ_MONTHLY` *[line 64]*

`SITE_MAP_FREQ_HOURLY = 'hourly'` *[line 58]*

`SITE_MAP_FREQ_MONTHLY = 'monthly'` *[line 61]*

`SITE_MAP_FREQ_NEVER = 'never'` *[line 63]*

`SITE_MAP_FREQ_WEEKLY = 'weekly'` *[line 60]*

`SITE_MAP_FREQ_YEARLY = 'yearly'` *[line 62]*

`SITE_MAP_PRIORITY_AVERAGE = '0.5'` *[line 69]*

`SITE_MAP_PRIORITY_HIGH = '0.75'` *[line 68]*

`SITE_MAP_PRIORITY_LOW = '0.25'` *[line 70]*

`SITE_MAP_PRIORITY_MAX = '1'` *[line 67]*

`SITE_MAP_PRIORITY_MIN = '0'` *[line 71]*

`SITE_MAP_SEARCH_ENGINE_MODE = false` *[line 54]*

**It will be for the search engines (sitemap. can be forgotten in that case.**

It will be for the search engines (sitemap.xml), all the pages which don't need to be present in the search engines results can be forgotten in that case.

`SITE_MAP_USER_MODE = true` *[line 49]*

**It will be a page of the site containing the site map**

It will be a page of the site containing the site map

# Class ModuleMap

[line 39]

The **ModuleMap** class represents the map of a module. It has a description (generally the module description) and contains some elements which can be some simple links or some sections (which can match the categories for example).

- **Package** content
- **Sub-Package** sitemap
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

## ModuleMap::\$description

*string* = [line 112]

- **Var** Description of the module

Constructor *void* function ModuleMap::ModuleMap(\$link) [line 45]

### Function Parameters:

- [SiteMapLink](#) **\$link** Link associated to the root of the module

## Builds a ModuleMap object

*void* function ModuleMap::export(&\$export\_config, \$export\_config) [line 84]

### Function Parameters:

- *SiteMapExportConfig* **\$export\_config** export configuration
- **&\$export\_config**

**Exports the sitemap (according to a configuration of templates). In your template, you will be able to use the following variables:**

- **MODULE\_NAME** which contains the name of the module

- **MODULE\_DESCRIPTION** which contains the description of the module
- **MODULE\_URL** which contains the URL of the module root page
- **DEPTH** which is the depth of the module map in the sitemap (generally 1).  
It might be useful to apply different CSS styles to each level of depth.
- **LINK\_CODE** which contains the code of the link associated to the module root exported with the same configuration.
- **C\_MODULE\_MAP** which is a boolean whose value is true, this will enable you to use a single template for the whole export configuration
- The loop "element" for which the variable **CODE** contains the code of each sub element of the module (for example categories)

*string* function ModuleMap::get\_description() [*line 55*]

**Return the module description**

*void* function ModuleMap::set\_description(\$description) [*line 64*]

**Function Parameters:**

- *string* **\$description** Description of the module

**Sets the description of the module**

## Class SiteMap

[*line 80*]

**Describes the map of the site. Can be exported according to any text form by using a template configuration. A site map contains some links, some link sections and some module maps (which also contain links and sections).**

- **Package** content
- **Sub-Package** sitemap
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

**SiteMap::\$elements**

*SiteMapElement[] = array()* [line 231]

- **Var** Elements contained by the site map

### SiteMap::\$site\_name

*string = "* [line 235]

- **Var** name of the site

Constructor *void* function SiteMap::SiteMap([\$site\_name = "], [\$elements = null]) [line 86]

#### **Function Parameters:**

- *SiteMapElement[]* **\$elements** List of the elements it contains
- **\$site\_name**

### Builds a SiteMap object with its elements

*void* function SiteMap::add(\$element) [line 125]

#### **Function Parameters:**

- [\*SiteMapElement\*](#) **\$element** The element to add

### Adds an element to the elements list of the SiteMap

*void* function SiteMap::build\_kernel\_map([\$mode = SITE\_MAP\_USER\_MODE], [\$auth\_mode = SITE\_MAP\_AUTH\_GUEST]) [line 179]

#### **Function Parameters:**

- **\$mode**
- **\$auth\_mode**

### Adds to the site map all the kernel links.

*void* function SiteMap::build\_modules\_maps() [*line 164*]

**Adds to the site map all maps of the installed modules**

*string* function SiteMap::export(&\$export\_config, \$export\_config) [*line 141*]

**Function Parameters:**

- *SiteMapExportConfig* **\$export\_config** Export configuration
- **&\$export\_config**

**Exports a SiteMap. You will be able to use the following variables into the templates used to export:**

- **C\_SITE\_MAP** which is a condition indicating if it's a site map (useful if you want to use a sigle template for the whole export configuration)
- **SITE\_NAME** which contains the name of the site
- A loop "element" in which the code of each element is in the variable **CODE**

*string* function SiteMap::get\_site\_name() [*line 99*]

**Returns the name of the site**

*void* function SiteMap::set\_site\_name(\$site\_name) [*line 108*]

**Function Parameters:**

- *string* **\$site\_name** name of the site

**Sets the name of the site. The default value is the name of the site taken from the site configuration.**

## Class SiteMapElement

[*line 38*]

**This abstract is the root of every object which can be contained by a SiteMap object. Some SiteMapElements objects can contain one or many SiteMapElement objects**

therefore the elements can be represented by a tree and each element has a depth in the tree.

- **Package** content
- **Sub-Package** sitemap
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >
- **Abstract Element**

### SiteMapElement::\$depth

*int = 1 [line 97]*

- **Var** Depth of the element in the elements tree

Constructor *void* function SiteMapElement::SiteMapElement(\$name) [line 44]

#### **Function Parameters:**

- *string* **\$name** Name of the object

### Builds a SiteMapElement object

*string* function SiteMapElement::export(&\$export\_config, \$export\_config, \$depth) [line 89]

#### **Function Parameters:**

- *SiteMapExportConfig* **\$export\_config** Export configuration
- *int* **\$depth** Depth of the element
- **&\$export\_config**

### Exports the element

*int* function SiteMapElement::get\_depth() [line 53]

### Returns the depth of the element in the tree

*string* function SiteMapElement::get\_name() [*line 62*]

**Returns the name of the menu**

*void* function SiteMapElement::set\_depth(\$depth) [*line 78*]

**Function Parameters:**

- *int* **\$depth** the depth of the element

**Sets the depth of the element**

## Class SitemapExportConfig

[*line 41*]

**Configuration used to export a SiteMap. It contains some Template objects which are used to export each kind of elements of a sitemap. Using different configurations will enable you for example to export in HTML code to be displayed in a page of the web site (the site map) or to be written in the sitemap.xml file at the root of your site, this file will be read by the search engines to optimize the research of your site.**

- **Package** content
- **Sub-Package** sitemap
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

**SitemapExportConfig::\$link\_file**

*Template* = [*line 149*]

- **Var** object used to export the SiteMapLink objects

**SitemapExportConfig::\$module\_map\_file**

*Template = [line 140]*

- **Var** object used to export the ModuleMap objects

### SitemapExportConfig::\$section\_file

*Template = [line 145]*

- **Var** object used to export the SiteMapSection objects

### SitemapExportConfig::\$site\_map\_file

*Template = [line 135]*

- **Var** object used to export the SiteMap objects

Constructor *void* function SitemapExportConfig::SitemapExportConfig(\$site\_map\_file, \$module\_map\_file, \$section\_file, \$link\_file) *[line 50]*

#### **Function Parameters:**

- *mixed* **\$module\_map\_file** The template used to export a ModuleMap object. Can be a Template object or a string (path of the template to use).
- *mixed* **\$section\_file** The template used to export a SiteMapSection object. Can be a Template object or a string (path of the template to use).
- *mixed* **\$link\_file** The template used to export a SiteMapLink object. Can be a Template object or a string (path of the template to use).
- **\$site\_map\_file**

### Builds a SiteMapExportConfig object

*Template* function SitemapExportConfig::get\_link\_stream() *[line 90]*

**Returns the Template object to use while exporting a SiteMapLink object.**

Template function SitemapExportConfig::get\_module\_map\_stream() [line 72]

**Returns the Template object to use while exporting a ModuleMap object.**

Template function SitemapExportConfig::get\_section\_stream() [line 81]

**Returns the Template object to use while exporting a SiteMapSection object.**

Template function SitemapExportConfig::get\_site\_map\_stream() [line 63]

**Returns the Template object to use while exporting a SiteMap object.**

void function SitemapExportConfig::set\_link\_stream(\$link\_file) [line 126]

**Function Parameters:**

- *mixed* **\$link\_file** The template used to export a SiteMapLink object. Can be a Template object or a string (path of the template to use).

**Sets the Template object to use while exporting a SiteMapLink object.**

void function SitemapExportConfig::set\_module\_map\_stream(\$module\_map\_file) [line 108]

**Function Parameters:**

- *mixed* **\$module\_map\_file** The template used to export a ModuleMap object. Can be a Template object or a string (path of the template to use).

**Sets the Template object to use while exporting a ModuleMap object.**

void function SitemapExportConfig::set\_section\_stream(\$section\_file) [line 117]

**Function Parameters:**

- *mixed* **\$section\_file** The template used to export a SiteMapSection object. Can be a Template object or a string (path of the template to use).

**Sets the Template object to use while exporting a SiteMapSection object.**

void function SitemapExportConfig::set\_site\_map\_stream(\$site\_map\_file, \$module\_map\_file) [line 99]

**Function Parameters:**

- *mixed* **\$module\_map\_file** The template used to export a Site object. Can be a Template object

or a string (path of the template to use).

- **\$site\_map\_file**

**Sets the Template object to use while exporting a Site object.**

## Class SiteMapLink

[line 39]

**This class represents a link of a site map.**

- **Package** content
- **Sub-Package** sitemap
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

### SiteMapLink::\$change\_freq

*string* = SITE\_MAP\_FREQ\_DEFAULT [line 246]

- **Var** Actualization frequency of the target page, must be a member of the following enum:  
SITE\_MAP\_FREQ\_ALWAYS, SITE\_MAP\_FREQ\_HOURLY, SITE\_MAP\_FREQ\_DAILY,  
SITE\_MAP\_FREQ\_WEEKLY, SITE\_MAP\_FREQ\_MONTHLY, SITE\_MAP\_FREQ\_YEARLY,  
SITE\_MAP\_FREQ\_NEVER, SITE\_MAP\_FREQ\_DEFAULT

### SiteMapLink::\$last\_modification\_date

*Date* = [line 251]

- **Var** Last modification date of the target page

### SiteMapLink::\$link

*Url = [line 240]*

- **Var** Url of the link

### SiteMapLink::\$name

*string = " [line 236]*

- **Var** Name of the SiteMapElement

### SiteMapLink::\$priority

*string = SITE\_MAP\_PRIORITY\_AVERAGE [line 256]*

- **Var** Priority of the target page, must be a member of the following enum:  
SITE\_MAP\_PRIORITY\_MAX, SITE\_MAP\_PRIORITY\_HIGH,  
SITE\_MAP\_PRIORITY\_AVERAGE, SITE\_MAP\_PRIORITY\_LOW, SITE\_MAP\_PRIORITY\_MIN

*string function SiteMapLink::export(&\$export\_config, \$export\_config) [line 212]*

#### **Function Parameters:**

- *SiteMapExportConfig* **\$export\_config** Export configuration
- **&\$export\_config**

**Exports the section according to the given configuration. You will use the following template variables:**

- **LOC** containing the URL of the link
- **TEXT** containing the name of the target page
- **C\_DISPLAY\_DATE** indicating if the date is not empty
- **DATE** containing the date of the last modification of the target page, formatted for the sitemap.xml file
- **ACTUALIZATION\_FREQUENCY** corresponding to the code needed in the sitemap.xml file

- **PRIORITY** corresponding to the code needed in the sitemap.xml file to indicate the priority of the target page.
- **C\_LINK** indicating that we are displaying a link (useful if you want to use a signe template export configuration)

*string* function SiteMapLink::get\_change\_freq() [*line 85*]

**Gets the change frequency (how often the target page is actualized)**

*Date* function SiteMapLink::get\_last\_modification\_date() [*line 104*]

**Returns the last modification date of the target page**

*Url* function SiteMapLink::get\_link() [*line 74*]

**Returns the URL of the link**

*string* function SiteMapLink::get\_name() [*line 65*]

**Returns the name of the target page**

*string* function SiteMapLink::get\_priority() [*line 95*]

**Gets the priority of the link**

*string* function SiteMapLink::get\_url() [*line 113*]

**Returns the URL of the link**

*void* function SiteMapLink::set\_change\_freq(\$change\_freq) [*line 156*]

**Function Parameters:**

- *string* **\$change\_freq** Frequency taken into the following enum: SITE\_MAP\_FREQ\_ALWAYS, SITE\_MAP\_FREQ\_HOURLY, SITE\_MAP\_FREQ\_DAILY, SITE\_MAP\_FREQ\_WEEKLY, SITE\_MAP\_FREQ\_MONTHLY, SITE\_MAP\_FREQ\_YEARLY, SITE\_MAP\_FREQ\_NEVER, SITE\_MAP\_FREQ\_DEFAULT

**Sets the change frequency**

*void* function SiteMapLink::set\_last\_modification\_date(\$last\_modification\_date, \$date) [*line 190*]

**Function Parameters:**

- *Date* **\$date** date
- **\$last\_modification\_date**

## Sets the last modification date of the target page

*void function SiteMapLink::set\_link(\$link) [line 138]*

### **Function Parameters:**

- [Uri](#) **\$link** URL

## Sets the URL of the link

*void function SiteMapLink::set\_name(\$name) [line 129]*

### **Function Parameters:**

- *string* **\$name** name of the element

## Sets the name of the element

*void function SiteMapLink::set\_priority(\$priority) [line 174]*

### **Function Parameters:**

- *string* **\$priority** Priority taken into the following enum: SITE\_MAP\_PRIORITY\_MAX, SITE\_MAP\_PRIORITY\_HIGH, SITE\_MAP\_PRIORITY\_AVERAGE, SITE\_MAP\_PRIORITY\_LOW, SITE\_MAP\_PRIORITY\_MIN

## Sets the priority of the link

*void function SiteMapLink::SitemapLink([\$name = "], [\$link = null], [\$change\_freq = SITE\_MAP\_FREQ\_MONTHLY], [\$priority = SITE\_MAP\_PRIORITY\_AVERAGE], [\$last\_modification\_date = null]) [line 52]*

### **Function Parameters:**

- *string* **\$name** Name of the target page
- [Uri](#) **\$link** link
- *string* **\$change\_freq** Frequency taken into the following enum: SITE\_MAP\_FREQ\_ALWAYS, SITE\_MAP\_FREQ\_HOURLY, SITE\_MAP\_FREQ\_DAILY, SITE\_MAP\_FREQ\_WEEKLY, SITE\_MAP\_FREQ\_MONTHLY, SITE\_MAP\_FREQ\_YEARLY, SITE\_MAP\_FREQ\_NEVER, SITE\_MAP\_FREQ\_DEFAULT
- *string* **\$priority** Priority taken into the following enum: SITE\_MAP\_PRIORITY\_MAX, SITE\_MAP\_PRIORITY\_HIGH, SITE\_MAP\_PRIORITY\_AVERAGE,

- `SITE_MAP_PRIORITY_LOW`, `SITE_MAP_PRIORITY_MIN`
- **`Date $last_modification_date`** Last modification date of the target page

**Builds a SiteMapLink object**

## Class SiteMapSection

*[line 37]*

**This class represents a section of a site map.**

- **Package** content
- **Sub-Package** sitemap
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

### SiteMapSection::\$elements

*SiteMapElement[] = array() [line 135]*

- **Var** List of the elements contained by the module map

### SiteMapSection::\$link

*SiteMapLink = [line 131]*

- **Var** Link associated to the section

Constructor *void* function SiteMapSection::SiteMapSection(\$link) *[line 43]*

**Function Parameters:**

- [SiteMapLink](#) **\$link** The link associated to the section

## Builds a SiteMapSection object

*void* function SiteMapSection::add(\$element) [*line 84*]

### Function Parameters:

- *SiteMapElement* **\$element** element to add

## Adds an element to the section

*string* function SiteMapSection::export(&\$export\_config, \$export\_config) [*line 105*]

### Function Parameters:

- *SiteMapExportConfig* **\$export\_config** Export configuration
- **&\$export\_config**

## Exports the section according to the given configuration. You will use the following template variables:

- **SECTION\_NAME** which contains the name of the section
- **SECTION\_URL** which contains the URL of the link associated to the section
- **DEPTH** which contains the depth of the section in the site map tree (useful for CSS classes names)
- **LINK\_CODE** which contains the code got by the associated link export
- **C\_SECTION**, boolean meaning that it's a section (useful if you want to use a single template for the whole export configuration)
- A loop "element" containing every element of the section (their code is available in the **CODE** variable of the loop)

*SiteMapLink* function SiteMapSection::get\_link() [*line 52*]

## Returns the link associated to the section

*void* function SiteMapSection::set\_depth(\$depth) [*line 70*]

### Function Parameters:

- *int* **\$depth** The depth of the element

## Sets the depth of the element

- **Warning** the description is not protected for XML displaying (but useless in sitemap.xml)

void function SiteMapSection::set\_link(\$link) [*line 61*]

### **Function Parameters:**

- [SiteMapLink](#) **\$link** the link

## Sets the link associated to the section

## atom.class.php

- **Package** content
- **Sub-Package** syndication

`DEFAULT_ATOM_TEMPLATE = 'framework/content/syndication/atom.tpl' [line 30]`

# feed.class.php

- **Package** content
- **Sub-Package** syndication

DEFAULT\_FEED\_NAME = 'master' *[line 29]*

ERROR\_GETTING\_CACHE = 'Error regenerating and / or retrieving the syndication cache of the %s (%s)' *[line 30]*

FEEDS\_PATH = PATH\_TO\_ROOT.'/cache/syndication/' *[line 28]*

## feed\_data.class.php

- **Package** content
- **Sub-Package** syndication

FEED\_DATA\_\_CLASS = 'FeedData' [*line 32*]

## rss.class.php

- **Package** content
- **Sub-Package** syndication

DEFAULT\_RSS\_TEMPLATE = 'framework/content/syndication/rss.tpl' *[line 28]*

## Class ATOM

*[line 42]*

This class could load a feed by its url or by a FeedData element and export it to the ATOM format

- **Package** content
- **Sub-Package** syndication
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >

Constructor *void* function ATOM::ATOM(\$module\_id, [\$feed\_name = DEFAULT\_FEED\_NAME], [\$id\_cat = 0]) *[line 51]*

### **Function Parameters:**

- *string* **\$module\_id** its module\_id
- *string* **\$feed\_name** the feeds name / type. default is DEFAULT\_FEED\_NAME
- *int* **\$id\_cat** the feed category id

### **Builds a new ATOM object**

*void* function ATOM::load\_file(\$url) *[line 61]*

### **Function Parameters:**

- *string* **\$url** the feed url

Loads a feed by its url

## Class Feed

[line 42]

This class could be used to export feeds

- **Package** content
- **Sub-Package** syndication
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >
- **Abstract Element** Do not use this class, but one of its children like RSS or ATOM

### Feed::\$data

*mixed* = null [line 174]

### Feed::\$id\_cat

*mixed* = 0 [line 170]

### Feed::\$module\_id

*mixed* = " [line 169]

### Feed::\$name

*mixed* = " [line 171]

### Feed::\$str

*mixed* = " [line 172]

### Feed::\$tpl

*mixed* = null [line 173]

Constructor *void* function Feed::Feed(\$module\_id, [\$name = DEFAULT\_FEED\_NAME], [\$id\_cat = 0]) [line 52]

#### **Function Parameters:**

- *string* **\$module\_id** its module\_id
- *string* **\$name** the feeds name / type. default is DEFAULT\_FEED\_NAME

- *int* **\$id\_cat** the feed category id

### Builds a new feed object

*void* function Feed::cache() [*line 150*]

#### Send the feed data in the cache

*void* function Feed::clear\_cache([\$module\_id = false]) [*line 186*]

#### Function Parameters:

- *mixed* **\$module\_id** the module module\_id or false. If false, Clear all feeds data from the cache

### Clear the cache of the specified module\_id.

- **Static**

*string* function Feed::export([\$template = false], [\$number = 10], [\$begin\_at = 0]) [*line 79*]

#### Function Parameters:

- *mixed* **\$template** If false, uses de default tpl. If an associative array, uses the default tpl but assigns it the array vars first. It could also be a Template object
- *int* **\$number** the number of item to display
- *int* **\$begin\_at** the first item to display

### Exports the feed as a string parsed by the <\$tpl> template

*string* function Feed::get\_cache\_file\_name() [*line 165*]

#### Returns the feed data cache filename

*string* function Feed::get\_feed\_menu(\$feed\_url) [*line 302*]

#### Function Parameters:

- *string* **\$feed\_url** Feed URL

Generates the code which shows all the feeds formats.

- **Static**

*string* function Feed::get\_parsed(\$module\_id, [\$name = DEFAULT\_FEED\_NAME], [\$idcat = 0], [\$tpl = false], [\$number = 10], [\$begin\_at = 0]) *[line 234]*

**Function Parameters:**

- *string* **\$module\_id** the module id
- *string* **\$name** the feed name / type
- *int* **\$idcat** the feed data category
- *mixed* **\$tpl** If false, uses de default tpl. If an associative array, uses the default tpl but assigns it the array vars first. It could also be a Template object
- *int* **\$number** the number of item to display
- *int* **\$begin\_at** the first item to display

## Export a feed

- **Static**

*bool* function Feed::is\_in\_cache() *[line 159]*

**Returns true if the feed data are in the cache**

*void* function Feed::load\_data(\$data) *[line 63]*

**Function Parameters:**

- [\*FeedData\*](#) **\$data** the element to load

## Loads a FeedData element

*void* function Feed::load\_file(\$url) *[line 68]*

### Function Parameters:

- *string* **\$url** the feed url

### Loads a feed by its url

*string* function Feed::read() [*line 133*]

### Loads the feed data in cache and export it

*void* function Feed::update\_cache(\$module\_id, \$name, &\$data, [\$idcat = 0], \$data) [*line 213*]

### Function Parameters:

- *string* **\$module\_id** the module id
- *string* **\$name** the feed name / type
- *&FeedData* **\$data** the data to put in the cache
- *int* **\$idcat** the feed data category
- **&\$data**

### Update the cache of the \$module\_id, \$name, \$idcat feed with \$data

- **Static**

## Class FeedData

[*line 40*]

### Contains meta-informations about a feed with its entries

- **Package** content
- **Sub-Package** syndication
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >

#### FeedData::\$auth\_bit

*mixed = 0 [line 170]*

#### FeedData::\$date

*mixed = null [line 165]*

#### FeedData::\$desc

*mixed = " [line 166]*

#### FeedData::\$host

*mixed = " [line 168]*

#### FeedData::\$items

*mixed = array() [line 169]*

#### FeedData::\$lang

*mixed = " [line 167]*

#### FeedData::\$link

*mixed = " [line 164]*

#### FeedData::\$title

*mixed = " [line 163]*

Constructor *void* function FeedData::FeedData([\$data = null]) [line 48]

#### **Function Parameters:**

- [FeedData](#) **\$data** an other FeedData object to clone

### **Builds a FeedData Object**

*void* function FeedData::add\_item(\$item) [line 106]

#### **Function Parameters:**

- **\$item**

*void* function FeedData::get\_date() [line 111]

*void* function FeedData::get\_date\_rfc822() [line 112]

*void* function FeedData::get\_date\_rfc3339() [*line 113*]  
*void* function FeedData::get\_desc() [*line 114*]  
*void* function FeedData::get\_host() [*line 116*]  
*FeedItem[]* function FeedData::get\_items() [*line 122*]

### Returns the feed items

*void* function FeedData::get\_lang() [*line 115*]  
*void* function FeedData::get\_link() [*line 110*]  
*void* function FeedData::get\_title() [*line 109*]  
*void* function FeedData::serialize() [*line 135*]  
*void* function FeedData::set\_auth\_bit(\$value) [*line 92*]

#### Function Parameters:

- *int* **\$value** the bit position in an int (from 1 to 32)

### Sets the feed auth bit, useful to check authorizations

*void* function FeedData::set\_date(\$value) [*line 72*]

#### Function Parameters:

- *Date* **\$value** a date object representing the feed date

### Sets the feed data date

*void* function FeedData::set\_desc(\$value) [*line 77*]

#### Function Parameters:

- *string* **\$value** the feed description

### Sets the feed description

*void* function FeedData::set\_host(\$value) [*line 87*]

#### Function Parameters:

- *string* **\$value** the feed host

### Sets the feed host

*void function FeedData::set\_lang(\$value) [line 82]*

**Function Parameters:**

- *string* **\$value** the feed language

### Sets the feed language

*void function FeedData::set\_link(\$value) [line 97]*

**Function Parameters:**

- *mixed* **\$value** a string url or an Url object

### Sets the feed item link

*void function FeedData::set\_title(\$value) [line 67]*

**Function Parameters:**

- *string* **\$value** The title

### Sets the feed title

*FeedItem[] function FeedData::subitems([\$number = 10], [\$begin\_at = 0]) [line 147]*

**Function Parameters:**

- *int* **\$number** the number of items to retrieve
- *int* **\$begin\_at** the number of the first to retrieve

Returns a items list containing \$number items starting from the \$begin\_at one

## Class FeedItem

[line 36]

## Contains meta-informations and informations about a feed entry / item

- **Package** content
- **Sub-Package** syndication
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com)>

### FeedItem::\$auth

*mixed* = null [line 117]

### FeedItem::\$date

*mixed* = null [line 113]

### FeedItem::\$desc

*mixed* = " [line 114]

### FeedItem::\$guid

*mixed* = " [line 115]

### FeedItem::\$image\_url

*mixed* = " [line 116]

### FeedItem::\$link

*mixed* = " [line 112]

### FeedItem::\$title

*mixed* = " [line 111]

Constructor *void* function FeedItem::FeedItem() [line 42]

### **Builds a FeedItem element**

*void* function FeedItem::get\_auth() [line 107]

*void* function FeedItem::get\_date() [line 102]

*void* function FeedItem::get\_date\_rfc822() [line 103]

*void* function FeedItem::get\_date\_rfc3339() [line 104]

*void* function FeedItem::get\_desc() [line 105]

*void* function FeedItem::get\_guid() [line 101]

*void* function FeedItem::get\_image\_url() [line 106]

*void* function FeedItem::get\_link() [line 100]

*void* function FeedItem::get\_title() [line 99]

*void* function FeedItem::set\_auth(\$auth, \$value) [line 69]

### **Function Parameters:**

- *int[string]* **\$value** the item authorizations array
- **\$auth**

### Sets the feed item auth, useful to check authorizations

*void function FeedItem::set\_date(\$value) [line 54]*

#### **Function Parameters:**

- *Date* **\$value** a date object representing the item date

### Sets the feed item date

*void function FeedItem::set\_desc(\$value) [line 59]*

#### **Function Parameters:**

- *string* **\$value** the feed item description

### Sets the feed item description

*void function FeedItem::set\_guid(\$value) [line 86]*

#### **Function Parameters:**

- *mixed* **\$value** a string url or an Url object

### Sets the feed item guid

*void function FeedItem::set\_image\_url(\$value) [line 64]*

#### **Function Parameters:**

- *string* **\$value** the picture url

### Sets the feed item picture

*void function FeedItem::set\_link(\$value) [line 74]*

### Function Parameters:

- *mixed* **\$value** a string url or an Url object

### Sets the feed item link

*void* function FeedItem::set\_title(\$value) [*line 49*]

### Function Parameters:

- *string* **\$value** The title

### Sets the feed item title

## Class FeedsCat

[*line 36*]

Describes a feed by building a category tree

- **Package** content
- **Sub-Package** syndication
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >

### FeedsCat::\$cat\_name

*mixed* = " [*line 109*]

### FeedsCat::\$children

*mixed* = array() [*line 111*]

### FeedsCat::\$id

*mixed* = 0 [*line 108*]

### FeedsCat::\$module\_id

*mixed = "* [line 110]

Constructor *void* function `FeedsCat::FeedsCat($module_id, $category_id, $category_name)` [line 44]

**Function Parameters:**

- *string* **\$module\_id** the feed module id
- *int* **\$category\_id** the category id
- *string* **\$category\_name** the category name

## Builds a FeedsCat Object

*void* function `FeedsCat::add_child($child)` [line 94]

**Function Parameters:**

- [FeedsCat](#) **\$child** The element to add

## Adds a FeedsCat child to the current FeedsCat object

*int* function `FeedsCat::get_category_id()` [line 76]

**Returns the category id**

*string* function `FeedsCat::get_category_name()` [line 85]

**Returns the category name**

*FeedsCat[]* function `FeedsCat::get_children()` [line 103]

**Returns the current category children**

*string* function `FeedsCat::get_module_id()` [line 66]

**Returns the module id**

*string* function `FeedsCat::get_url([$feed_type = ""])` [line 56]

**Function Parameters:**

- *string* **\$feed\_type** The feed type

## Returns the feed url

# Class FeedsList

[line 37]

This class contains an agregation of differents feeds

- **Package** content
- **Sub-Package** syndication
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com)>

## FeedsList::\$list

*array = array()* [line 63]

- **Var** Contains the feeds trees

Constructor *void* function FeedsList::FeedsList() [line 39]

*void* function FeedsList::add\_feed(\$cat\_tree, \$feed\_type) [line 46]

### **Function Parameters:**

- [FeedsCat](#) **\$cat\_tree** The feed to add to the list
- *string* **\$feed\_type** The feed category name

**Adds the FeedsCat \$cat\_tree to the current feeds list**

*FeedsCat[]* function FeedsList::get\_feeds\_list() [line 55]

**Returns the feeds list**

# Class RSS

[line 40]

This class could load a feed by its url or by a FeedData element and export it to the RSS format

- **Package** content
- **Sub-Package** syndication
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com)>

Constructor *void* function RSS::RSS(\$module\_id, [\$feed\_name = DEFAULT\_FEED\_NAME], [\$id\_cat = 0]) [line 49]

**Function Parameters:**

- *string* **\$module\_id** its module\_id
- *string* **\$feed\_name** the feeds name / type. default is DEFAULT\_FEED\_NAME
- *int* **\$id\_cat** the feed category id

## Builds a new RSS object

*void* function RSS::load\_file(\$url) [line 59]

**Function Parameters:**

- *string* **\$url** the feed url

## Loads a feed by its url



# Package core Procedural Elements

## application.class.php

- **Package** core

```
APPLICATION_TYPE__KERNEL = 'kernel' [line 28]  
APPLICATION_TYPE__MODULE = 'module' [line 29]  
APPLICATION_TYPE__TEMPLATE = 'template' [line 30]
```

# cache.class.php

- **Package** core

```
NO_FATAL_ERROR_CACHE = true [line 29]  
RELOAD_CACHE = true [line 28]
```

# errors.class.php

- **Package** core

```
ARCHIVE_ALL_ERRORS = true [line 30]  
ARCHIVE_ERROR = true [line 31]  
DISPLAY_ALL_ERROR = false [line 35]  
E_STRICT = 2048 [line 38]  
NO_ARCHIVE_ERROR = false [line 32]  
NO_FILE_ERROR = " [line 34]  
NO_LINE_ERROR = " [line 33]
```

# menu\_service.class.php

- **Package** core

MOVE\_DOWN = 1 [*line 37*]  
MOVE\_UP = -1 [*line 36*]

# stats\_saver.class.php

- **Package** core

include\_once **PATH\_TO\_ROOT.'/lang/'.\$CONFIG['lang'].'/stats.php'** *[line 29]*

# updates.class.php

- **Package** core

```
CHECK_ALL_UPDATES = CHECK_KERNEL|CHECK_MODULES|CHECK_THEMES [line 35]
CHECK_KERNEL = 0X01 [line 32]
CHECK_MODULES = 0X02 [line 33]
CHECK_THEMES = 0X04 [line 34]
PHPBOOST_OFFICIAL_REPOSITORY = 'http://www.phpboost.com/repository/main.xml' [line 29]
PHP_MIN_VERSION_UPDATES = '5' [line 30]
```

# Package core Classes

## Class Application *[line 40]*

- **Package** core
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >

### Application::\$authors

*mixed = array() [line 373]*

### Application::\$bug\_corrections

*mixed = array() [line 378]*

### Application::\$compatibility\_max

*mixed = " [line 365]*

### Application::\$compatibility\_min

*mixed = " [line 364]*

### Application::\$description

*mixed = " [line 375]*

### Application::\$download\_url

*mixed = " [line 370]*

### Application::\$id

*mixed = " [line 355]*

### Application::\$improvements

*mixed = array()* [line 377]

#### **Application::\$language**

*mixed = "* [line 357]

#### **Application::\$localized\_language**

*mixed = "* [line 358]

#### **Application::\$name**

*mixed = "* [line 356]

#### **Application::\$new\_features**

*mixed = array()* [line 376]

#### **Application::\$priority**

*mixed = null* [line 367]

#### **Application::\$pubdate**

*mixed = null* [line 366]

#### **Application::\$repository**

*mixed = "* [line 361]

#### **Application::\$security\_improvements**

*mixed = array()* [line 379]

#### **Application::\$security\_update**

*mixed = false* [line 368]

#### **Application::\$type**

*mixed = "* [line 359]

#### **Application::\$update\_url**

*mixed = "* [line 371]

#### **Application::\$version**

*mixed = "* [line 363]

#### **Application::\$warning**

*mixed = null* [line 382]

#### **Application::\$warning\_level**

*mixed = null* [line 381]

Constructor *void* function Application::Application(\$id, \$language, [\$type = APPLICATION\_TYPE\_\_MODULE],

*[\$version = 0], [\$repository = "]] [line 50]*

**Function Parameters:**

- ***\$id* \$id**
- ***\$language* \$language**
- ***\$type* \$type** see APPLICATION\_TYPE\_XXX constants
- ***\$version* \$version**
- ***\$repository* \$repository**

## constructor of the class

*boolean* function Application::check\_compatibility() *[line 175]*

### **Checks compatibility with limits**

*void* function Application::get\_authors() *[line 283]*

### **Accessor of Authors**

*void* function Application::get\_bug\_corrections() *[line 299]*

### **Accessor of Bug Corrections Text**

*void* function Application::get\_compatibility\_max() *[line 259]*

### **Accessor of Compatibility Max value**

*void* function Application::get\_compatibility\_min() *[line 255]*

### **Accessor of Compatibility min value**

*void* function Application::get\_description() *[line 287]*

### **Accessor of Description Text**

*void* function Application::get\_download\_url() *[line 275]*

### **Accessor of Download URL**

*void* function Application::get\_id() *[line 227]*

### **Accessor of id**

*string* function Application::get\_identifier() *[line 167]*

### **Gets an identifier of the application**

*void* function Application::get\_improvements() *[line 295]*

## Accessor of Improvments Text

*void function Application::get\_language() [line 235]*

### Accessor of Language

*void function Application::get\_localized\_language() [line 239]*

### Accessor of Localized language

*void function Application::get\_name() [line 231]*

### Accessor of name

*void function Application::get\_new\_features() [line 291]*

### Accessor of New Features Text

*void function Application::get\_priority() [line 267]*

### Accessor of Priority

*void function Application::get\_pubdate() [line 263]*

### Accessor of Publication Date

*void function Application::get\_repository() [line 247]*

### Accessor of Repository

*void function Application::get\_security\_improvments() [line 303]*

### Accessor of Security Improvments

*void function Application::get\_security\_update() [line 271]*

### Accessor of Security Update

*void function Application::get\_type() [line 243]*

### Accessor of Type

*void function Application::get\_update\_url() [line 279]*

### Accessor of Update URL

*void function Application::get\_version() [line 251]*

### Accessor of Version

void function Application::get\_warning() [line 311]

### Accessor of Warning

void function Application::get\_warning\_level() [line 307]

### Accessor of Warning level

void function Application::load(&\$xml\_desc) [line 67]

#### Function Parameters:

- *\$xml\_desc* &**\$xml\_desc** reference to xml description

### Loads an XML description

## Class BreadCrumb

[line 36]

This class is used to represent the bread crumb displayed on each page of the website. It enables the user to locate himself in the whole site. A bread crumb can look like this: Home >> My module >> First level category >> Second level category >> Third level category >> .. >> My page >> Edition

- **Package** core
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

### BreadCrumb::\$array\_links

*string* = array() [line 122]

- **Var** List of the links

Constructor *void* function BreadCrumb::BreadCrumb() [*line 41*]

**Builds a BreadCrumb object.**

*void* function BreadCrumb::add(\$text, [\$target = ""]) [*line 50*]

**Function Parameters:**

- *string* **\$text** Name of the page
- *string* **\$target** Link whose target is the page

**Adds a link in the bread crumb. This link will be put at the end of the list.**

*void* function BreadCrumb::clean() [*line 113*]

**Removes all the existing links.**

*void* function BreadCrumb::display() [*line 83*]

**Displays the bread crumb.**

*void* function BreadCrumb::remove\_last() [*line 75*]

**Removes the last link of the list**

*void* function BreadCrumb::reverse() [*line 67*]

**Reverses the whole list of the links. It's very useful when it's easier for you to make the list in the reverse way, at the end, you only need to reverse the list and it will be ok.**

## Class Cache

[*line 39*]

**This class is the cache manager of PHPBoost. Its functioning is very particular. Loading a file is equivalent to include the file. The cache file must define some PHP global variables. They will be usable in the execution context of the page. You should read on the PHPBoost website the documentation which explains you how to integrate a cache for you module, it's too much complex to be explained here.**

- **Package** core
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

## Cache::\$files

*string[]* = array('config', 'debug', 'modules', 'menus', 'htaccess', 'themes', 'langs', 'css', 'day', 'groups', 'member', 'uploads', 'com', 'ranks', 'writingpad', 'smileys', 'stats') [*line 658*]

- **Var** List of all the cache files of the kernel.
- **Static**

Constructor *void* function Cache::Cache() [*line 44*]

**Builds a Cache object. Check if the directory in which the cache is written is writable.**

*true* function Cache::delete\_file(\$file) [*line 199*]

**Function Parameters:**

- *string* **\$file** Name of the file to delete.

**Deletes a cache file.**

*void* function Cache::generate\_all\_files() [*line 164*]

**Regenerates all the cache files managed by the PHPBoost cache manager. This method needs a lot of resource, call it only when you are sure you need it.**

*void* function Cache::generate\_all\_modules() [*line 178*]

**Generates all the module cache files.**

*void* function Cache::generate\_file(\$file) [*line 133*]

**Function Parameters:**

- *string* **\$file** The name of the file to generate.

**Generates a file according to the specified method.**

*void function Cache::generate\_module\_file(\$module\_name, [\$no\_alert\_on\_error = false]) [line 143]*

**Function Parameters:**

- *string* **\$module\_name** Name of the module for which you want to generate the cache.
- *bool* **\$no\_alert\_on\_error** true if you want to display the generation error, false otherwise.

### Generates a module file

*void function Cache::load(\$file, [\$reload\_cache = false]) [line 61]*

**Function Parameters:**

- *string* **\$file** Identifier of the cache file (for example the name of your module).
- *bool* **\$reload\_cache** If the cache file may have been already loaded, RELOAD\_CACHE force it to be reloaded, for example if the file has been updated since the first loading.

### Loads a file file.

*void function Cache::write(\$module\_name, &\$cache\_string, \$cache\_string) [line 216]*

**Function Parameters:**

- *string* **\$module\_name** Name of the file to write
- *string* **\$cache\_string** Content of the file to write
- **&\$cache\_string**

### Writes a cache file.

## Class Errors

*[line 45]*

**This class is the error manager of PHPBoost. It is designed to collect and store all errors occurs in the projet.**

- **Package** core
- **Author** Viarre Régis crowkait@phpboost.com

#### Errors::\$archive\_all

*mixed = [line 414]*

#### Errors::\$personal\_tpl

*mixed = false [line 417]*

#### Errors::\$redirect

*mixed = [line 415]*

#### Errors::\$template

*mixed = [line 416]*

Constructor *void* function Errors::Errors([\$archive\_all = false]) *[line 52]*

#### **Function Parameters:**

- *boolean* **\$archive\_all** TRUE archive all events FALSE if not

### constructor

*string* function Errors::display(\$errstr, \$errno, [\$errline = "], [\$errfile = "], [\$archive = false]) *[line 243]*

#### **Function Parameters:**

- *string* **\$errstr** The text which explain the error.
- *int* **\$errno** The error type (use the PHP errors constants).
- *string* **\$errline** The error line (use the constant `__LINE__`).
- *string* **\$errfile** The file where the error is located (use the constant `__FILE__`).
- *boolean* **\$archive** (optional) Backup the error in the error.log file

### Exception handler for developer, return the error.

*void* function Errors::get\_erno\_class(\$errno) *[line 346]*

#### **Function Parameters:**

- **\$errno**

## Get Error type

*void function Errors::get\_last\_\_error\_log() [line 315]*

### Get last error informations

*void function Errors::handler(\$errstr, \$errno, [\$errline = ""], [\$errfile = ""], [\$tpl\_cond = ""], [\$archive = false], [\$stop = true]) [line 151]*

#### Function Parameters:

- *string* **\$errstr** The text which explain the error.
- *int* **\$errno** The error type (use the PHP errors constants).
- *string* **\$errline** The error line (use the constant `__LINE__`).
- *string* **\$errfile** The file where the error is located (use the constant `__FILE__`).
- *string* **\$tpl\_cond** (optional) This argument allow you to display error in a template condition.
- *boolean* **\$archive** (optional) Backup the error in the error.log file
- *boolean* **\$stop** Avoid redirect loop.

## Exception handler for developer.

*void function Errors::handler\_php(\$errno, \$errstr, \$errfile, \$errline) [line 81]*

#### Function Parameters:

- *string* **\$errno** error number
- *string* **\$errstr** error label
- *string* **\$errfile** file name
- *string* **\$errline** line number

## PHP exceptions handler

*void function Errors::set\_default\_template() [line 304]*

### Set default template for the handler methods.

*void function Errors::set\_template(&\$template) [line 295]*

#### Function Parameters:

- **&\$template**

## Set a personal template for the handler methods.

# Class MenuService

[line 46]

This service manage kernel menus by adding the persistance to menus objects. It also provides all moving and disabling methods to change the website appearance.

- **Package** core
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >
- **Static**

*bool* function MenuService::add\_mini\_menu(\$menu, &\$installed\_menus\_names) [line 396]

**Function Parameters:**

- *Folder* **\$menu** the menu folder
- **&\$installed\_menus\_names**

## Add the menu named in the \$menu folder

*bool* function MenuService::add\_mini\_module(\$module, [\$generate\_cache = true]) [line 490]

**Function Parameters:**

- *string* **\$module** the module name
- **\$generate\_cache**

## Add the module named \$module mini modules

*void* function MenuService::assign\_positions\_conditions(&\$template, \$position) [line 678]

**Function Parameters:**

- *&Template* **&\$template** the template to use
- *int* **\$position** the menu position

## Assigns the positions conditions for different printing modes

*void function MenuService::change\_position(&\$menu, [\$direction = MOVE\_UP], \$menu, \$diff) [line 269]*

### **Function Parameters:**

- *Menu* **\$menu** The menu to move
- *int* **\$diff** the direction to move it. positives integers move down, negatives, up.
- **&\$menu**
- **\$direction**

## Change the menu position in a block

*void function MenuService::delete(&\$menu, \$menu) [line 191]*

### **Function Parameters:**

- *mixed* **\$menu** The (Menu) Menu or its (int) id to delete from the database
- **&\$menu**

## Delete a Menu from the database

*void function MenuService::delete\_mini\_menu(\$menu) [line 427]*

### **Function Parameters:**

- *string* **\$menu** the mini menu name

## delete the mini menu named \$menu

*void function MenuService::delete\_mini\_module(\$module) [line 542]*

### **Function Parameters:**

- *string* **\$module** the mini module name

## delete the mini module \$module

*void function MenuService::delete\_module\_feeds\_menus(\$module\_id) [line 612]*

***Function Parameters:***

- *string* **\$module\_id** the module id

## Delete all the feeds menus with the this module id

*void function MenuService::disable(&\$menu, \$menu) [line 219]*

***Function Parameters:***

- [\*Menu\*](#) **\$menu** the menu to disable
- **&\$menu**

## Disable a menu

*void function MenuService::enable(&\$menu, \$menu) [line 209]*

***Function Parameters:***

- [\*Menu\*](#) **\$menu** the menu to enable
- **&\$menu**

## Enable a menu

*void function MenuService::enable\_all([\$enable = true]) [line 323]*

***Function Parameters:***

- *bool* **\$enable** if true enables all menus otherwise, disables them

## Enables or disables all menus

*void function MenuService::generate\_cache([\$return\_string = false]) [line 345]*

***Function Parameters:***

- **\$return\_string**

## Generate the cache

*unknown\_type* function MenuService::get\_menus\_map() [*line 87*]

*unknown\_type* function MenuService::get\_menu\_list([\$class = MENU\_\_CLASS], [\$block = BLOCK\_POSITION\_\_ALL], [\$enabled = MENU\_ENABLE\_OR\_NOT]) [*line 55*]

### **Function Parameters:**

- *\$block* **\$class**
- *\$enabled* **\$block**
- **\$enabled**

*Menu* function MenuService::load(\$id) [*line 123*]

### **Function Parameters:**

- *int* **\$id** the id of the Menu to retrieve from the database

## Retrieve a Menu Object from the database by its id

*void* function MenuService::move(&\$menu, \$block, [\$save = true], \$menu) [*line 231*]

### **Function Parameters:**

- *Menu* **\$menu** the menu to move
- *int* **\$block** the destination block
- *bool* **\$save** if true, save also the menu
- **&\$menu**

## Move a menu into a block and save it. Enable or disable it according to the destination block

*bool* function MenuService::save(&\$menu, \$menu) [*line 141*]

### **Function Parameters:**

- *Menu* **\$menu** The Menu to save
- **&\$menu**

## save a Menu in the database

*int* function MenuService::str\_to\_location(\$str\_location) [*line 701*]

**Function Parameters:**

- *string* **\$str\_location** the location

## Convert the string location the int location

*void* function MenuService::update\_mini\_menus\_list([\$update\_cache = true]) [*line 444*]

**Function Parameters:**

- *bool* **\$update\_cache** if true it will also regenerate the cache

## Update the mini menus list by adding new ones and delete old ones

*void* function MenuService::update\_mini\_modules\_list([\$update\_cache = true]) [*line 559*]

**Function Parameters:**

- *bool* **\$update\_cache** if true it will also regenerate the cache

## Update the mini modules list by adding new ones and delete old ones

*LinksMenu* function MenuService::website\_modules([\$menu\_type = VERTICAL\_MENU]) [*line 629*]

**Function Parameters:**

- *int* **\$menu\_type** the menu type

## Return a menu with links to modules

# Class Repository

[*line 35*]

- **Package** core
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com)>

### Repository::\$url

*mixed = " [line 99]*

### Repository::\$xml

*mixed = null [line 100]*

Constructor *void* function Repository::Repository(\$url) *[line 41]*  
**Function Parameters:**

- **\$url \$url**

### constructor of the class

*void* function Repository::check(\$app) *[line 58]*  
**Function Parameters:**

- **\$app \$app**

### Check Application

*void* function Repository::get\_url() *[line 97]*  
**Accessor of url**

## Class StatsSaver

*[line 36]*

- **Package** core
- **Author** Viarre Régis [crowkait@phpboost.com](mailto:crowkait@phpboost.com)

*void function StatsSaver::compute\_referer() [line 42]*

### **Compute Stats of Site Referers**

*void function StatsSaver::compute\_users() [line 104]*

### **Compute Stats of Site Users**

## **Class Updates**

*[line 45]*

- **Package** core
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >

### **Updates::\$apps**

*mixed = array() [line 170]*

### **Updates::\$repositories**

*mixed = array() [line 169]*

Constructor *void function Updates::Updates([\$checks = CHECK\_ALL\_UPDATES]) [line 51]*

#### **Function Parameters:**

- ***\$checks*** ***\$checks***

### **constructor of the class**



# Package db Procedural Elements

## mysql.class.php

- **Package db**

```
CONNECTED_TO_DATABASE = 3 [line 36]  
CONNECTION_FAILED = 1 [line 34]  
DBTYPE = 'mysql' [line 38]  
DB_NO_CONNECT = false [line 29]  
ERRORS_MANAGEMENT_BY_RETURN = false [line 30]  
EXPLICIT_ERRORS_MANAGEMENT = true [line 31]  
LOW_PRIORITY = 'LOW_PRIORITY' [line 28]  
UNEXISTING_DATABASE = 2 [line 35]
```

# Package db Classes

## Class Backup

[line 33]

This class helps you to generate the backup file of your data base.

- **Package** db
- **Author** Benoît Sautel ben.popeye@gmail.com / Régis Viarre crowkait@phpboost.com

### Backup::\$backup\_script

*string* = " [line 276]

- **Var** Backup script

### Backup::\$tables

*string[]* = array() [line 272]

- **Var** List of the tables used by PHPBoost.

Constructor *void* function Backup::Backup() [line 38]

**Builds a Backup object**

void function Backup::concatenate\_to\_query(\$string) [*line 155*]

**Function Parameters:**

- *string* **\$string** String to concatenate.

**Concatenates a string at the end of the current script.**

void function Backup::export\_file(\$file\_path) [*line 217*]

**Function Parameters:**

- *string* **\$file\_path** Path of the file.

**Writes the backup script in a text file.**

unknown\_type function Backup::extract\_table\_structure([\$tables = array()]) [*line 231*]

**Function Parameters:**

- *\$tables* **\$tables**

void function Backup::generate\_create\_table\_query([\$table\_list = array()]) [*line 83*]

**Function Parameters:**

- *string[]* **\$table\_list** names of the tables which must be created by the backup script. If you want to generate the query which will create all the tables, don't use this parameter or let an empty array.

**Concatenates the tables creation to the SQL backup script.**

void function Backup::generate\_drop\_table\_query([\$table\_list = array()]) [*line 65*]

**Function Parameters:**

- *string[]* **\$table\_list** names of the tables which must be dropped by the query. If you want to generate the query which will drop all the tables, don't use this parameter or let an empty array.

**Concatenates the query which drops the PHPBoost tables only if they exist to the backup SQL script.**

*void* function Backup::generate\_insert\_values\_query([\$tables = array()]) [*line 107*]

**Function Parameters:**

- **\$tables** **\$tables** names of the tables which must be filled by the backup script. If you want to generate the query which will fill all the tables, don't use this parameter or let an empty array.

**Concatenates the tables content insertion queries to the SQL backup script.**

*string* function Backup::get\_script() [*line 164*]

**Returns the current backup script.**

*string[]* function Backup::get\_tables\_list() [*line 197*]

**Retrieves the list of the tables used by PHPBoost.**

*int* function Backup::get\_tables\_number() [*line 207*]

**Returns the number of tables used by PHPBoost.**

*string[]* function Backup::get\_tables\_properties\_list() [*line 187*]

**Lists the tables (name and informations relative to each table) of the data base at which is connected this SQL object. This method calls the SHOW TABLE STATUS MySQL query, to know more about it, see <http://dev.mysql.com/doc/refman/5.1/en/show-table-status.html>**

*void* function Backup::list\_db\_tables() [*line 49*]

**Retrieves the list of the tables present on the database used.**

## Class Sql

[*line 55*]

**This class manages all the database access done by PHPBoost. It currently manages only one DBMS, MySQL, but we made it as generic as we could. It doesn't support ORM (Object Relational Mapping). On PHPBoost, all the table which are used contain a prefix which enables for example to install several instances of the software on the same data base. When you execute a query in a table, concatenate**

the `PREFIX` constant before the name of your table. Notice also that the kernel tables can have their name changed. You must not use their name directly but the constants which are defined in the file `/kernel/db/tables.php`.

If you encounter any problem when writing queries, you should search what you need in the MySQL documentation, which is very well done: <http://dev.mysql.com/doc/>

- **Package** db
- **Author** Régis Viarre [crowkait@phpboost.com](mailto:crowkait@phpboost.com), Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >

#### Sql::\$base\_name

```
string = " [line 736]
```

- **Var** name of the data base at which is connected the object.

#### Sql::\$connected

```
bool = false [line 732]
```

- **Var** true if the object is connected to the data base, false otherwise.

#### Sql::\$link

```
resource = [line 724]
```

- **Var** Link with the data base

#### Sql::\$req

```
int = 0 [line 728]
```

- **Var** Number of requests executed by the object.

Constructor *void* function `Sql::Sql()` [*line 60*]

**Builds a MySQL connection.**

*int* function `Sql::affected_rows($resource, [$query = "])` [*line 340*]

**Function Parameters:**

- *resource* **\$resource** Resource corresponding to the request. The resource is given by the method which execute some queries in the data base.
- *string* **\$query** Deprecated field. Don't use it.

**Returns the number of the rows which have been affected by a request.**

*void* function `Sql::auto_connect()` [*line 129*]

**Connects automatically the application to the DBMS by reading the database configuration file whose path is `/kernel/db/config.php`. If an error occurs while connecting to the server, the script execution will be stopped and the error will be written in the page.**

The function `Sql::clean_database_name($db_name)` [*line 714*]

**Function Parameters:**

- *string* **\$db\_name** Name to clear

**Cleans the data base name to be sure it's a correct name**  
Cleans the data base name to be sure it's a correct name

- **Static**

*bool* function `Sql::close()` [*line 392*]

## Closes the current MySQL connection if it is open.

*string* function `Sql::concat($param)` [line 297]

### Function Parameters:

- *string* **\$param** Element to concatenate. Repeat this argument for each element you want to concatenate, in the same order.

Generates the syntax to use the concatenation operator (CONCAT in MySQL). The MySQL fields names must be in a PHP string for instance between simple quotes: 'field\_name' The PHP variables must be bordered by simple quotes, for example: "\ . \$my\_var . \"

*int* function `Sql::connect($sql_host, $sql_login, $sql_pass, $base_name, [$errors_management = EXPLICIT_ERRORS_MANAGEMENT])` [line 84]

### Function Parameters:

- *string* **\$sql\_host** Name or IP address of the server on which is the DBMS you want to use.
- *string* **\$sql\_login** Login enabling PHPBoost to connect itself to the DBMS (the MySQL login).
- *string* **\$sql\_pass** Password enabling PHPBoost to connect itself to the DBMS.
- *\$base\_name* **\$base\_name** string Name of the data base PHPBoost must join an work on.
- *\$errors\_management* **\$errors\_management** bool The way according to which you want to manage the data base connection errors :
  - `ERRORS_MANAGEMENT_BY_RETURN` will return the error
  - `EXPLICIT_ERRORS_MANAGEMENT` will stop the script execution and display the error message

This method enables you to connect to the DBMS when you have the data base access informations.

*int* function `Sql::count_table($table, $errline, $errfile)` [line 268]

### Function Parameters:

- *string* **\$table** Table name
- *int* **\$errline** The number of the line at which you call this method. Use the `__LINE__` constant. It is very interesting when you debug your script and you want to know where is called the query which returns an error.
- *int* **\$errfile** The file in which you call this method. Use the `__FILE__` constant. It is very interesting when you debug your script and you want to know where is called the query which returns an error.

**Counts the number of the row contained in a table.**

*string* function `Sql::create_database($db_name)` [*line 616*]

**Function Parameters:**

- *string* **\$db\_name** Name of the data base to create

**Creates a data base on the DBMS at which is connected the current object.**

*string* function `Sql::date_diff($field)` [*line 372*]

**Function Parameters:**

- *string* **\$field** Name of the field against which you want to compute the number of years.

**Generates the MySQL syntax which enables you to compute the number of years separating a date in a data base field and today.**

*void* function `Sql::drop_tables($table_array)` [*line 682*]

**Function Parameters:**

- *string[]* **\$table\_array** List of the tables to drop.

**Drops some tables in the data base.**

*string* function `Sql::escape($value)` [*line 628*]

**Function Parameters:**

- *string* **\$value** String to escape

**Escapes the dangerous characters in the string you inject in your requests.**

*string[]* function `Sql::fetch_assoc($result)` [*line 316*]

**Function Parameters:**

- *resource* **\$result** MySQL result resource to browse. The resource is provided by the `query_while` method.

**Browses a MySQL result resource row per row. When you call this method on a resource, you get the next row.**

*string[]* function `Sql::fetch_row($result)` [line 328]

**Function Parameters:**

- *resource* **\$result** MySQL result resource to browse. The resource is provided by the `query_while` method.

**Browses a MySQL result resource row per row. When you call this method on a resource, you get the next row.**

*string* function `Sql::get_data_base_name()` [line 589]

**Returns the name of the data base which with the object is connected.**

*string* function `Sql::get_dbms_version()` [line 580]

**Gets the version of MySQL used.**

*int* function `Sql::get_executed_requests_number()` [line 512]

**Returns the number of request executed by this object.**

*string* function `Sql::highlight_query($query)` [line 523]

**Function Parameters:**

- *string* **\$query** Query to highlight

**Highlights a SQL query to be more readable by a human.**

- **Static**

*string* function `Sql::indent_query($query)` [*line 556*]

**Function Parameters:**

- *string* **\$query** Query to indent.

**Indents a MySQL query.**

- **Static**

*int* function `Sql::insert_id([$query = "])` [*line 362*]

**Function Parameters:**

- *\$query* **\$query**

**Gets the ID generated from the previous INSERT operation.**

*string* function `Sql::limit($start, [$num_lines = 0])` [*line 284*]

**Function Parameters:**

- *int* **\$start** Number of the first row (0 is the first).
- *int* **\$num\_lines** Number of the rows you want to retrieve.

**Builds the MySQL syntax used to impose a limit in your row selection.**

*string[]* function `Sql::list_databases()` [*line 599*]

**Lists the existing data bases on the DBMS at which the object is connected. Only the data bases visible for the user connected will be returned.**

*string[]* function `Sql::list_fields($table)` [*line 410*]

**Function Parameters:**

- *string* **\$table** Name of the table.

**Lists all the columns of a table.**

*string[]* function `Sql::list_tables()` [*line 445*]

**Lists the tables (name and informations relative to each table) of the data base at which is connected this SQL object. This method calls the SHOW TABLE STATUS MySQL query, to know more about it, see <http://dev.mysql.com/doc/refman/5.1/en/show-table-status.html>**

*int* function `Sql::num_rows($resource, $query)` [*line 351*]

**Function Parameters:**

- *resource* **\$resource** Resource corresponding to the result of the query.
- *\$query* **\$query** Deprecated field. Don't use it.

**Returns the number of rows got by a selection query.**

*void* function `Sql::optimize_tables($table_array)` [*line 648*]

**Function Parameters:**

- *string[]* **\$table\_array** List of the tables to optimize.

**Optimizes some tables in the data base.**

*void* function `Sql::parse($file_path, [$tableprefix = "])` [*line 474*]

**Function Parameters:**

- *string* **\$file\_path** Path of the file.
- *string* **\$tableprefix** prefix The prefix you want to work with.

**Parses a SQL file. The SQL file contains the name of the tables with the prefix phpboost\_.**

*string* function `Sql::query($query, $errline, $errfile)` [*line 156*]

**Function Parameters:**

- *string* **\$query** Selection query
- *int* **\$errline** The number of the line at which you call this method. Use the `__LINE__` constant. It

is very interesting when you debug your script and you want to know where is called the query which returns an error.

- *int* **\$errfile** The file in which you call this method. Use the `__FILE__` constant. It is very interesting when you debug your script and you want to know where is called the query which returns an error.

**Sends a simple selection query to the DBMS and retrieves the result. A simple query selects only one field in one row.**

*void* function `Sql::query_array($table, $field, $clause, $errline, $errfile)` [line 186]

**Function Parameters:**

- *string* **\$table** Name of the table in which you want to select the values
- *string* **\$field** Name of the field for which you want to retrieve the value. If you want to work on several fields, you have to repeat this parameter for each field you want to select.
- *string* **\$clause** Where clause which will enable the method to know in which row it must select the values. It must respect the MySQL syntax and start off with 'WHERE '.
- *int* **\$errline** The number of the line at which you call this method. Use the `__LINE__` constant. It is very interesting when you debug your script and you want to know where is called the query which returns an error.
- *int* **\$errfile** The file in which you call this method. Use the `__FILE__` constant. It is very interesting when you debug your script and you want to know where is called the query which returns an error.

**This method makes automatically a query on several fields of a row. You tell it in which table you want to select, which row you want to use, and it will return you the values. It takes a variable number of parameters.**

*bool* function `Sql::query_close($resource)` [line 382]

**Function Parameters:**

- *resource* **\$resource** Resource you want to deallocate.

**Frees the memory allocated for a resource.**

*resource* function `Sql::query_inject($query, $errline, $errfile)` [line 234]

**Function Parameters:**

- *string* **\$query** The query you want to execute
- *int* **\$errline** The number of the line at which you call this method. Use the `__LINE__` constant. It is very interesting when you debug your script and you want to know where is called the query

which returns an error.

- *int* **\$errfile** The file in which you call this method. Use the `__FILE__` constant. It is very interesting when you debug your script and you want to know where is called the query which returns an error.

**This method enables you to execute CUD (Create Update Delete) queries in the database, and more generally, any query which has not any return value.**

*resource* function `Sql::query_while($query, $errline, $errfile)` [line 251]

**Function Parameters:**

- *int* **\$errline** The number of the line at which you call this method. Use the `__LINE__` constant. It is very interesting when you debug your script and you want to know where is called the query which returns an error.
- *int* **\$errfile** The file in which you call this method. Use the `__FILE__` constant. It is very interesting when you debug your script and you want to know where is called the query which returns an error.
- *\$query* **\$query** The query you want to execute

**This method enables you to execute a Retrieve query on several rows in the data base.**

*void* function `Sql::repair_tables($table_array)` [line 660]

**Function Parameters:**

- *string[]* **\$table\_array** List of the tables to repair.

**Repairs some tables in the data base.**

*void* function `Sql::truncate_tables($table_array)` [line 672]

**Function Parameters:**

- *string[]* **\$table\_array** List of the tables to truncate.

**Trucates some tables in the data base.**



# Package events Procedural Elements

## administrator\_alert.class.php

- **Package** events

ADMIN\_ALERT\_HIGH\_PRIORITY = 4 *[line 39]*

ADMIN\_ALERT\_LOW\_PRIORITY = 2 *[line 35]*

ADMIN\_ALERT\_MEDIUM\_PRIORITY = 3 *[line 37]*

ADMIN\_ALERT\_STATUS\_PROCESSED = EVENT\_STATUS\_PROCESSED *[line 47]*

ADMIN\_ALERT\_STATUS\_UNREAD = EVENT\_STATUS\_UNREAD *[line 45]*

ADMIN\_ALERT\_VERY\_HIGH\_PRIORITY = 5 *[line 41]*

ADMIN\_ALERT\_VERY\_LOW\_PRIORITY = 1 *[line 33]*

# administrator\_alert\_service.class.php

- **Package** events

ADMINISTRATOR\_ALERT\_TYPE = 1 [*line 31*]

# contribution.class.php

- **Package** events

CONTRIBUTION\_AUTH\_BIT = 1 [*line 34*]

# contribution\_service.class.php

- **Package** events

CONTRIBUTION\_TYPE = 0 [*line 31*]

# event.class.php

- **Package** events

```
EVENT_STATUS_BEING_PROCESSED = 1 [line 35]  
EVENT_STATUS_PROCESSED = 2 [line 37]  
EVENT_STATUS_UNREAD = 0 [line 33]
```

# Package events Classes

## Class AdministratorAlert

[line 56]

This class represents an alert which must be sent to the administrator. It allows to the module developers to handle the administrator alerts. The administrator alerts can be in the administration panel and can be used when you want to signal an important event to the administrator(s).

- **Package** events
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

### AdministratorAlert::\$priority

*int* = ADMIN\_ALERT\_MEDIUM\_PRIORITY [line 184]

- **Var** Priority of the alert
- **Access** protected

### AdministratorAlert::\$properties

*string* = " [line 190]

- **Var** Properties of the alert (string field of unlimited length) which can for example contain a serializes array of object.

- **Access** protected

Constructor *void* function AdministratorAlert::AdministratorAlert() [*line 61*]

**Builds an AdministratorAlert object.**

*void* function AdministratorAlert::build(\$id, \$entitled, \$properties, \$fixing\_url, \$current\_status, \$creation\_date, \$id\_in\_module, \$identifier, \$type, \$priority) [*line 81*]

**Function Parameters:**

- *int* **\$id** Identifier of the alert.
- *string* **\$entitled** Entitled of the alert.
- *string* **\$properties** Properties of the alert.
- *string* **\$fixing\_url** Fixing url.
- *int* **\$current\_status** Alert status.
- *Date* **\$creation\_date** Alert creation date?
- *int* **\$id\_in\_module** Id in module field.
- *string* **\$identifier** Identifier of the alert.
- *string* **\$type** Type of the alert.
- *int* **\$priority** Priority of the alert.

**Builds an alert from its whole parameters.**

*int* function AdministratorAlert::get\_priority() [*line 99*]

**Gets the priority of the alert.**

*string* function AdministratorAlert::get\_priority\_name() [*line 154*]

**Gets the priority name. It's automatically translated to the user language, ready to be displayed.**

*string* function AdministratorAlert::get\_properties() [*line 108*]

**Gets the alert properties.**

*void* function AdministratorAlert::set\_priority(\$priority) [*line 124*]

**Function Parameters:**

- *int* **\$priority** The priority, it must be one of those values:
  - ADMIN\_ALERT\_VERY\_LOW\_PRIORITY Very low priority
  - ADMIN\_ALERT\_LOW\_PRIORITY Low priority
  - ADMIN\_ALERT\_MEDIUM\_PRIORITY Medium priority
  - ADMIN\_ALERT\_HIGH\_PRIORITY High priority
  - ADMIN\_ALERT\_VERY\_HIGH\_PRIORITY Very high priority

**Sets the priority of the alert.**

*void function AdministratorAlert::set\_properties(\$properties) [line 141]*

**Function Parameters:**

- *string* **\$properties** Properties.

**Sets the properties of the alert.**

## Class AdministratorAlertService

*[line 39]*

**This static class allows you to handler easily the administrator alerts which can be made in PHPBoost.**

- **Package** events
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >
- **Static**

*int[] function AdministratorAlertService::compute\_number\_unread\_alerts() [line 251]*

**Counts the number of unread alerts.**

- **Static**

*void function AdministratorAlertService::delete\_alert(&\$alert, \$alert) [line 228]*

**Function Parameters:**

- [AdministratorAlert](#) **\$alert** The alert to delete.
- **&\$alert**

**Deletes an alert from the database.**

*AdministratorAlert[]* function AdministratorAlertService::find\_by\_criteria([\$id\_in\_module = null], [\$type = null], [\$identifier = null]) [*line 81*]

**Function Parameters:**

- *int* **\$id\_in\_module** Id in the module.
- *string* **\$type** Alert type.
- *string* **\$identifier** Alert identifier.

**Builds a list of alerts matching the required criteria(s). You can specify many criterias. When you use several of them, it's a AND condition. It will only return the alert which match all the criterias.**

- **Static**

*AdministratorAlert* function AdministratorAlertService::find\_by\_id(\$alert\_id) [*line 47*]

**Function Parameters:**

- *int* **\$alert\_id** Id of the alert.

**Builds an alert knowing its id.**

- **Static**

*AdministratorAlert[]* function AdministratorAlertService::find\_by\_identifier(\$identifier, [\$type = ""]) [*line 133*]

**Function Parameters:**

- *string* **\$identifier** The identifier of the alerts you look for.
- *string* **\$type** The type of the alert you look for.

**Finds an alert knowing its identifier and maybe its type.**

- **Static**

*AdministratorAlerts[]* function AdministratorAlertService::get\_all\_alerts([\$criteria = 'creation\_date'], [\$order = 'desc'], [\$begin = 0], [\$number = 20]) [*line 165*]

**Function Parameters:**

- *string* **\$criteria** The criteria according to which you want to order. It can be id, entitled, fixing\_url, current\_status, creation\_date, identifier, id\_in\_module, type, priority, description.
- *string* **\$order** asc or desc.
- *int* **\$begin** You want all the alert from the (\$begin+1)(th).
- *int* **\$number** The number of alerts you want.

**Lists all the alerts of the site. You can order them. You can also choose how much alerts you want.**

- **Static**

*int* function AdministratorAlertService::get\_number\_alerts() [*line 276*]

**Returns the number of alerts.**

- **Static**

*int* function AdministratorAlertService::get\_number\_unread\_alerts() [*line 265*]

Returns the number of unread alerts.

- **Static**

*void function AdministratorAlertService::save\_alert(&\$alert, \$alert) [line 194]*

**Function Parameters:**

- [AdministratorAlert](#) **\$alert** The alert to create or update.
- **&\$alert**

Create or updates an alert in the database. It creates it whether it doesn't exist or updates it if it already exists.

- **Static**

## Class Contribution

*[line 42]*

This class represents a contribution made by a user to complete the content of the website. All the contributions are managed in the contribution panel.

- **Package** events
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

**Contribution::\$auth**

*array = array() [line 329]*

- **Var** Authorization array containing the people who can treat the contribution.
- **Access** protected

#### **Contribution::\$description**

*string = [line 311]*

- **Var** Description of the contribution (for instance to justify a contribution).
- **Access** protected

#### **Contribution::\$fixer\_id**

*int = 0 [line 341]*

- **Var** Identifier of the member who has fixed the contribution.
- **Access** protected

#### **Contribution::\$fixer\_login**

*string = " [line 353]*

- **Var** Login of the member who has fixed the contribution.
- **Access** protected

#### **Contribution::\$fixing\_date**

*Date = [line 323]*

- **Var** Date at which the contribution has been processed (if it is obviously). Default value: date at which is created the contribution.

- **Access** protected

#### Contribution::\$module

*string = " [line 317]*

- **Var** String containing the identifier of the module corresponding to the contribution (ex: forum).
- **Access** protected

#### Contribution::\$poster\_id

*int = 0 [line 335]*

- **Var** Identifier of the member who has posted the contribution.
- **Access** protected

#### Contribution::\$poster\_login

*string = " [line 347]*

- **Var** Login of the member who has posted the contribution.
- **Access** protected

Constructor *void* function Contribution::Contribution() *[line 47]*

#### **Builds a Contribution object.**

*void* function Contribution::build(\$id, \$entitled, \$description, \$fixing\_url, \$module, \$status, \$creation\_date, \$fixing\_date, \$auth, \$poster\_id, \$fixer\_id, \$id\_in\_module, \$identifier, \$type, [\$poster\_login = "], [\$fixer\_login = "])  
*[line 74]*

#### **Function Parameters:**

- *int* **\$id** Contribution id.
- *string* **\$entitled** Contribution entitled.
- *string* **\$fixing\_url** URL associated to the event.
- *int* **\$module** status Contribution status.

- ***Date* \$creation\_date** Contribution creation date.
- ***Date* \$fixing\_date** Contribution fixing date.
- ***mixed[]* \$auth** Auth array determining the people who can treat the contribution.
- ***int* \$poster\_id** Contribution creator id.
- ***int* \$fixer\_id** Contribution fixer id.
- ***int* \$id\_in\_module** Id of the element associated to the contribution.
- ***string* \$identifier** Contribution identifier.
- ***string* \$type** Contribution type.
- ***string* \$poster\_login** Login of the poster of the contribution.
- ***string* \$fixer\_login** Login of the fixer of the contribution.
- **\$description**
- **\$status**

**Builds a contribution object from its whole parameters.**

*mixed[]* function Contribution::get\_auth() [*line 228*]

**Gets the authorization of treatment of this contribution.**

*string* function Contribution::get\_description() [*line 201*]

**Gets the description of the contribution.**

*int* function Contribution::get\_fixer\_id() [*line 246*]

**Gets the identifier of the fixer.**

*string* function Contribution::get\_fixer\_login() [*line 264*]

**Gets the fixer login.**

The function Contribution::get\_fixing\_date() [*line 219*]

**Gets the contribution fixing date.**

*string* function Contribution::get\_module() [*line 210*]

**Gets the module in which the contribution is used.**

*string* function Contribution::get\_module\_name() [*line 292*]

**Gets the name of the module in which the contribution is used.**

*int* function Contribution::get\_poster\_id() [*line 237*]

**Gets the identifier of the poster.**

*string* function Contribution::get\_poster\_login() [*line 255*]

**Gets the poster login.**

*string* function Contribution::get\_status\_name() [*line 273*]

**Gets the contribution status name. It's automatically translated in the user language, ready to be displayed.**

*void* function Contribution::set\_auth(\$auth) [*line 149*]

**Function Parameters:**

- *mixed*[] **\$auth** Auth array.

**Sets the authorization of the contribution. It will determine who will be able to treat the contribution.**

*void* function Contribution::set\_description(\$description) [*line 191*]

**Function Parameters:**

- *string* **\$description** Description (can be some HTML content).

**Sets the description of the contribution.**

*void* function Contribution::set\_fixer\_id(\$fixer\_id) [*line 175*]

**Function Parameters:**

- *int* **\$fixer\_id** Id.

**Sets the id of the fixer.**

*void* function Contribution::set\_fixing\_date(\$date) [*line 106*]

**Function Parameters:**

- *Date* **\$date** Date

**Sets the fixing date.**

*void* function Contribution::set\_module(\$module) [*line 97*]

### Function Parameters:

- *string* **\$module** Module identifier (for example the name of the module folder).

**Sets the module in which the contribution is used.**

*void* function Contribution::set\_poster\_id(\$poster\_id) [*line 159*]

### Function Parameters:

- *int* **\$poster\_id** Id.

**Sets the id of the poster.**

*void* function Contribution::set\_status(\$new\_current\_status) [*line 121*]

### Function Parameters:

- *int* **\$new\_current\_status** One of those elements:
  - EVENT\_STATUS\_UNREAD if it's not read.
  - EVENT\_STATUS\_BEING\_PROCESSED if the event is beeing processed
  - EVENT\_STATUS\_PROCESSED if the event is processed.

**Set the status of the contribution.**

## Class ContributionService

[*line 39*]

**This service allows developers to manage their contributions.**

- **Package** events
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >
- **Static**

*int[]* function ContributionService::compute\_number\_contrib\_for\_each\_profile() [*line 239*]

**Computes the number of contributions available for each profile. It will count the contributions for the administrator, the moderators, the members, for each group and for each member who can have some special authorizations.**

- **Static**

*void* function ContributionService::delete\_contribution(&\$contribution, \$contribution) [*line 200*]

**Function Parameters:**

- [\*Contribution\*](#) **\$contribution** The contribution to delete in the data base.
- **&\$contribution**

**Deletes a contribution in the database.**

*Contribution[]* function ContributionService::find\_by\_criteria(\$module, [\$id\_in\_module = null], [\$type = null], [\$identifier = null], [\$poster\_id = null], [\$fixer\_id = null]) [*line 117*]

**Function Parameters:**

- *string* **\$module** The module identifier.
- *int* **\$id\_in\_module** The id in module field.
- *string* **\$type** The contribution type.
- *string* **\$identifier** The contribution identifier.
- *int* **\$poster\_id** The poster.
- *int* **\$fixer\_id** The fixer.

**Builds a list of the contributions matching the required criteria(s). All the parameters represent the criterias you can use. If you don't want to use a criteria, let the null value. The returned contribution match all the criterias (it's a AND condition).**

- **Static**

*Contribution* function `ContributionService::find_by_id($id_contrib)` [*line 47*]

**Function Parameters:**

- *int* **\$id\_contrib** Id of the contribution.

**Finds a contribution with its identifier.**

- **Static**

*void* function `ContributionService::generate_cache()` [*line 220*]

**Generates the contribution cache file.**

- **Static**

*Contribution[]* function `ContributionService::get_all_contributions([$criteria = 'creation_date'], [$order = 'desc'])` [*line 79*]

**Function Parameters:**

- *string* **\$criteria** Criteria according to which they are ordered. It can be id, entitled, fixing\_url, auth, current\_status, module, creation\_date, fixing\_date, poster\_id, fixer\_id, poster\_member.login poster\_login, fixer\_member.login fixer\_login, identifier, id\_in\_module, type, description.
- *string* **\$order** desc or asc.

**Gets all the contributions of the table. You can sort the list.**

- **Static**

`void function ContributionService::save_contribution(&$contribution, $contribution) [line 167]`

**Function Parameters:**

- [Contribution](#) **\$contribution** The contribution to synchronize with the data base.
- **&\$contribution**

**Create or update a contribution in the database.**

- **Static**

## Class Event

[line 50]

**It's the common part between two types of event existing now in PHPBoost:**

- **User contribution managed into the contribution panel**
- **Administrator alert, triggered for example when a new update is available or when a new member account is to approbate**

- **Package events**
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >
- **Abstract Element**

**Event::\$creation\_date**

*Date = [line 319]*

- **Var** The event creation date.

- **Access** protected

#### Event::\$current\_status

*int = EVENT\_STATUS\_UNREAD [line 313]*

- **Var** The event status.
- **Access** protected

#### Event::\$entitled

*string = " [line 301]*

- **Var** Entitled (title or name) of the event.
- **Access** protected

#### Event::\$fixing\_url

*string = " [line 307]*

- **Var** URL where you can process the event (relative from the website root).
- **Access** protected

#### Event::\$id

*int = 0 [line 295]*

- **Var** Numerical identifier of the event (in DB).
- **Access** protected

#### Event::\$identifier

*string = " [line 332]*

- **Var** Identifier to recognize the entry (optionnal).
- **Access** protected

#### **Event::\$id\_in\_module**

*int = 0 [line 326]*

- **Var** Id corresponding to the event in the module (optionnal).
- **Access** protected

#### **Event::\$must\_regenerate\_cache**

*bool = true [line 344]*

- **Var** To know if the modifications implies to regenerate the cache (for instance whether the status has been changed).
- **Access** protected

#### **Event::\$type**

*string = " [line 338]*

- **Var** Event type (optionnal).
- **Access** protected

Constructor *void* function Event::Event() [line 55]

#### **Builds an Event object.**

*void* function Event::build(\$id, \$entitled, \$fixing\_url, \$current\_status, \$creation\_date, \$id\_in\_module, \$identifier, \$type) [line 277]

#### **Function Parameters:**

- *int* **\$id** The event id.
- *string* **\$entitled** The event entitled.
- *string* **\$fixing\_url** The URL corresponding to the event.
- *int* **\$current\_status** The event status.
- *Date* **\$creation\_date** The creation date.
- *int* **\$id\_in\_module** The id of the object associated to the event.
- *string* **\$identifier** The event identifier.
- *string* **\$type** The event type.

**Builds an event object from its whole parameters.**

*Date* function `Event::get_creation_date()` [*line 205*]

**Returns the creation date of the event.**

*string* function `Event::get_entitled()` [*line 173*]

**Returns the entitled of the event.**

*string* function `Event::get_fixing_url()` [*line 182*]

**Returns the URL corresponding to the alert.**

*int* function `Event::get_id()` [*line 164*]

**Gets the id of the event (in the event data base).**

*string* function `Event::get_identifier()` [*line 224*]

**Gets the identifier of the event. To retrieve your event, you might need to have a field in which you put some informations, for example a hash or an identifier. It's that identifier which can be used to filter the events.**

*int* function `Event::get_id_in_module()` [*line 214*]

**Gets the id in the module. This value corresponds to the id of the daba base entry associated to the event.**

*bool* function `Event::get_must_regenerate_cache()` [*line 242*]

**Gets the value indicating if the cache must be generated.**

*int* function `Event::get_status()` [*line 196*]

**Gets the status of the event. The status is one of those elements: ul>  
<li>EVENT\_STATUS\_UNREAD if it's not read.</li>  
<li>EVENT\_STATUS\_BEING\_PROCESSED if the event is beeing**

**processed** **EVENT\_STATUS\_PROCESSED** if the event is processed.

The function `Event::get_status_name()` [line 251]

**Gets the event status name. It's automatically translated in the user language.**

*string* function `Event::get_type()` [line 233]

**Gets the type of the event. To retrieve your event, you might need to have a type of event, for example if your module has different kinds of events. With this field, you can specify it.**

*void* function `Event::set_creation_date($date)` [line 115]

**Function Parameters:**

- ***Date* \$date** The creation date

**Sets the creation date of the event.**

*void* function `Event::set_entitled($entitled)` [line 75]

**Function Parameters:**

- ***string* \$entitled** The event entitled.

**Sets the entitled of the event. The entitled can be considered as the name, it must be explicit.**

*void* function `Event::set_fixing_url($fixing_url)` [line 84]

**Function Parameters:**

- ***string* \$fixing\_url** Relative URL (the first character must be / for the root of the site).

**Sets the URL corresponding to the event. For the contributions and the administrator alerts it's the number URL at which the problem can be solved.**

*void* function `Event::set_id($id)` [line 65]

**Function Parameters:**

- *int* **\$id** Id of the event.

**Sets the id of the event. The id is the corresponding data base entry one.**

*void* function Event::set\_identifier(\$identifier) [*line 136*]

**Function Parameters:**

- *string* **\$identifier** Identifier of the event.

**Sets the event identifier. To retrieve your event, you might need to have a field in which you put some informations, for example a hash or an identifier. It's that identifier which can be used to filter the events. You don't have to use it, you can let it blank.**

*void* function Event::set\_id\_in\_module(\$id) [*line 126*]

**Function Parameters:**

- *int* **\$id** Id in the module

**Sets the id in module parameter. It corresponds to the id of the element corresponding to the event in your data base tables. For example, if you use the events to allow user to purpose some news in your web site, it will be the id of the news added.**

*void* function Event::set\_must\_regenerate\_cache(\$must) [*line 154*]

**Function Parameters:**

- *bool* **\$must** true if we must generate the events cache, otherwise false.

**Sets a private property indicating if the changes made on this event imply the regeneration of the events cache.**

*void* function Event::set\_status(\$new\_current\_status) [*line 98*]

**Function Parameters:**

- *int* **\$new\_current\_status** One of those elements:
  - EVENT\_STATUS\_UNREAD if it's not read.
  - EVENT\_STATUS\_BEING\_PROCESSED if the event is beeing processed
  - EVENT\_STATUS\_PROCESSED if the event is processed.

## Set the status of the event.

`void function Event::set_type($type) [line 145]`

### **Function Parameters:**

- *string* **\$type** The type of your event.

**Sets the type of the event. To retrieve your event, you might need to have a type of event, for example if your module has different kinds of events. With this field, you can specify it.**



# Package io Procedural Elements

## mail.class.php

- **Package** io

`CRLF = "\r\n" [line 28]`

`MIME_FORMAT_HTML = 'text/html' [line 30]`

`MIME_FORMAT_TEXT = 'text/plain' [line 29]`

# template.class.php

- **Package** io

```
AUTO_LOAD_FREQUENT_VARS = true [line 31]  
DO_NOT_AUTO_LOAD_FREQUENT_VARS = false [line 32]  
TEMPLATE_STRING_MODE = true [line 29]  
TEMPLATE_WRITE_MODE = false [line 30]
```

# upload.class.php

- **Package** io

```
CHECK_EXIST = true [line 32]  
DELETE_ON_ERROR = true [line 29]  
NO_DELETE_ON_ERROR = false [line 30]  
NO_UNIQ_NAME = false [line 33]  
UNIQ_NAME = true [line 31]
```

# Package io Classes

## Class Mail

[line 38]

This class allows you to send mails without having to deal with the mail headers and parameters.

- **Package** io
- **Author** RÃ©gis Viarre <[crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

### Mail::\$content

*string* = " [line 304]

- **Var** content of the mail

### Mail::\$format

*string* = MIME\_FORMAT\_TEXT [line 329]

- **Var** MIME of the mail content

### Mail::\$headers

*The = " [line 319]*

- **Var** mail headers.

### **Mail::\$object**

*sting = " [line 299]*

- **Var** object of the mail

### **Mail::\$recipients**

*string[] = array() [line 324]*

- **Var** Recipients of the mail. If they are more than one, a comma separates their addresses.

### **Mail::\$sender\_mail**

*string = " [line 309]*

- **Var** Address of the mail sender.

### **Mail::\$sender\_name**

*string = " [line 314]*

- **Var** The mail sender name.

Constructor *void* function Mail::Mail() [line 43]

**Builds a Mail object.**

*bool* function Mail::check\_validity(\$mail\_address) [line 249]

**Function Parameters:**

- **\$mail\_address**

**Checks that an email address has a correct form.**

- **Static**

*string* function Mail::get\_content() [line 167]

**Returns the mail content.**

*string* function Mail::get\_headers() [line 176]

**Returns the mail headers.**

*string* function Mail::get\_mime() [line 194]

**Returns the MIME type of the mail content**

Returns the MIME type of the mail content

*string* function Mail::get\_object() [line 158]

**Returns the mail object.**

*string* function Mail::get\_recipients() [line 149]

**Returns the mail recipients' addresses. They are separated by a comma.**

*string* function Mail::get\_sender\_mail() [line 131]

**Returns the mail address of the sender.**

*string* function Mail::get\_sender\_name() [line 140]

**Returns the mail sender's name.**

*bool* function Mail::send() [*line 233*]

**Sends the mail.**

*bool* function Mail::send\_from\_properties(\$mail\_to, \$mail\_object, \$mail\_content, \$mail\_from, [\$mail\_header = null], [\$sender\_name = 'admin']) [*line 210*]

**Function Parameters:**

- *string* **\$mail\_to** The mail recipients' address.
- *string* **\$mail\_object** The mail object.
- *string* **\$mail\_content** content of the mail
- *string* **\$mail\_from** The mail sender's address.
- *string* **\$mail\_header** The header you want to specify (it you don't specify it, it will be generated automatically).
- *string* **\$sender\_name** The mail sender's name. If you don't use this parameter, the name of the site administrator will be taken.

**Sends the mail.**

- **Deprecated**

*void* function Mail::set\_content(\$content) [*line 113*]

**Function Parameters:**

- *string* **\$content** The mail content

**The mail content.**

*void* function Mail::set\_headers(\$headers) [*line 122*]

**Function Parameters:**

- *string* **\$headers** The mail headers.

**Sets the headers. Forces them, they won't be generated automatically.**

*void function Mail::set\_mime(\$mime) [line 185]*

**Function Parameters:**

- *string* **\$mime** MIME\_FORMAT\_TEXT or MIME\_FORMAT\_HTML

### Sets the MIME type of the mail content

Sets the MIME type of the mail content

*void function Mail::set\_object(\$object) [line 104]*

**Function Parameters:**

- *string* **\$object** Mail object

### Sets the mail object

*void function Mail::set\_recipients(\$recipients) [line 73]*

**Function Parameters:**

- *string* **\$recipients** Recipients of the mail. If they are more than one, use the comma to separate their addresses.

### Sets the recipient(s) of the mail.

*bool function Mail::set\_sender(\$sender, [\$sender\_name = 'admin']) [line 53]*

**Function Parameters:**

- *string* **\$sender** The mail sender address.
- *string* **\$sender\_name** 'admin' if the mail is sent by the administrator, 'user' otherwise.

### Sets the mail sender.

*void function Mail::\_generate\_headers() [line 260]*

### Generates the mail headers.

- **Access** protected

## Class Template

*[line 80]*

This class allows you to handle a template file. Your template files should have the .tpl extension. **The PHPBoost template syntax** **Simple variables** A simple variable is accessible with the {NAME} syntax where NAME is its template name. If the variable is not assigned, nothing will be displayed (no error message). Simple variables are assigned by the assign\_vars() method. **Loops** You can make some loops to repeat a pattern, those loops can be nested. A loop has a name (name) and each iteration contains some variables, for example, the variable VAR.

- **START name #**  
My variable is {name.VAR}
- **END name #**  
To nest loops, here is an example:
- **START loop1 #**  
I write my loop1 var here: {loop1.VAR}.
- **START loop1.loop2 #**  
I can write my loop2 var here: {loop1.loop2.VAR} but also my loop1 var of the parent loop: {loop1.VAR}.

- **END loop1 #**  
To assign the variable, see the assign\_block\_vars() method which creates one iteration. **Conditions** When you want to display something only in particular case, you can use some condition tests.

- **IF C\_MY\_TEST #**  
This text will be displayed only if the C\_MY\_TEST variable is true
- **ENDIF #**

You can nest some conditions.

To be more efficient, this class uses a cache and parses each file only once. **File paths** The web site can have several themes whose files aren't in the same folders. When you load a file, you just have to load the generic file and the good template file will be loaded dynamically. **Kernel template file** When you want to load a kernel template file, the path you must indicate is only the name of the file, for example header.tpl loads /template/your\_theme/header.tpl and if it doesn't exist, it will load /template/default/header.tpl. **Module template file** When you want to load a module template file, you must indicate the name of you module and then the name of the file like this: module/file.tpl which

will load the `/module/templates/file.tpl`. If the user themes redefines the `file.tpl` file for the module `module`, the file `templates/your_theme/modules/module/file.tpl` will be loaded. 

## Menu template file

 To load a file of a menu, use this kind of path: `menus/my_menu/file.tpl` which will load the `/menus/my_menu/templates/file.tpl` file. 

## Framework template file

 To load a framework file, use a path like this: `framework/package/file.tpl` which will load `/templates/your_theme/framework/package/file.tpl` if the theme overrides it, otherwise `/templates/default/framework/package/file.tpl` will be used.

- **Package** io
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) > Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com) >

### Template::\$files

```
string[] = array() [line 745]
```

- **Var** Map `file_identfier => file_path`. The paths have been checked before to be entered in this map.

### Template::\$module\_data\_path

```
string[] = array() [line 750]
```

- **Var** Map `module_name => module_data_path`

### Template::\$return\_mode

```
bool = [line 755]
```

- **Var** Parsing type of the current parsing operation. True if it's in string mode, false in write mode.

## Template::\$template

*string* = " [line 740]

- **Var** Content of the cache file corresponding to a loaded file just before it to be cached.

## Template::\$tpl

*string* = " [line 735]

- **Var** Template file name

Constructor *void* function Template::Template([*\$tpl* = ""], [*\$auto\_load\_vars* = AUTO\_LOAD\_FREQUENT\_VARS]) [line 88]

### **Function Parameters:**

- *string* **\$tpl** Path of your TPL file. See the class description to know you you have to write this path.
- *bool* **\$auto\_load\_vars** AUTO\_LOAD\_FREQUENT\_VARS if you want to assign the frequent vars (user lang, user theme, CSRF token, current path to root, user logged in or not...). DO\_NOT\_AUTO\_LOAD\_FREQUENT\_VARS if you don't want.

## **Builds a Template object.**

*void* function Template::assign\_block\_vars(*\$block\_name*, *\$array\_vars*) [line 163]

### **Function Parameters:**

- *string* **\$block\_name** Block name.
- *string[]* **\$array\_vars** A map *var\_name* => *var\_value*. Generally, *var\_name* is written in caps characters.

**Assigns a template block. A block represents a loop and has a name which be used in your template file to indicate which loop you want to browse. To know what syntax to use to browse a loop, see the class description, there are examples.**

*void* function `Template::assign_vars($array_vars)` [*line 149*]

**Function Parameters:**

- *string[]* **\$array\_vars** A map `var_name => var_value`. Generally, `var_name` is written in caps characters.

**Assigns some simple template vars. Those variables will be accessed in your template with the `{var_name}` syntax.**

*Template* function `Template::copy()` [*line 265*]

**Clones this object.**

*string* function `Template::get_module_data_path($module)` [*line 136*]

**Function Parameters:**

- *string* **\$module** Name of the module for which you want to know the data path.

**Retrieves the path of the module. This path will be used to write the relative paths in your templates.**

*string* function `Template::parse([$return_mode = TEMPLATE_WRITE_MODE])` [*line 202*]

**Function Parameters:**

- *bool* **\$return\_mode** In its default behaviour (`TEMPLATE_WRITE_MODE`), this class write what it parses in the PHP standard output. If you want to retrieve the parsed content and not to write it, use the `TEMPLATE_STRING_MODE` variable.

**Parses the file. It will use the variables you assigned.**

*string* function `Template::pparse($parse_name, [$return_mode = false])` [*line 222*]

**Function Parameters:**

- *string* **\$parse\_name** The identifier of the file you want to parse.
- *bool* **\$return\_mode** In its default behaviour (`TEMPLATE_WRITE_MODE`), this class write what it parses in the PHP standard output.

**Parses the file whose name is `$parse_name` and which has been declared with the**

**set\_filenames\_method.** It uses the variables you assigned (when you assign a variable it will be usable in every file handled by this object).

- **Deprecated**

*void* function Template::set\_filenames(\$array\_tpl) [*line 118*]

**Function Parameters:**

- *string[]* **\$array\_tpl** A map file\_identifier => file\_path where file\_identifier is the name you give to your file and file\_path its path. See the class description to learn how to write the path.

**Loads several files in the same Template instance.**

- **Deprecated**

*void* function Template::unassign\_block\_vars(\$block\_name) [*line 188*]

**Function Parameters:**

- *string* **\$block\_name** The name of your block.

**Deletes a block. It won't be browsable any more in your template.**

*bool* function Template::\_check\_cache\_file(\$tpl\_path, \$file\_cache\_path) [*line 422*]

**Function Parameters:**

- *string* **\$tpl\_path** File path to control.
- *string* **\$file\_cache\_path** Cache file path corresponding.

**Allows you to know if the cache file is still valid or if it has to be removed because the source file has been updated.**

- **Access** protected

*string* function `Template::_check_file($filename)` [line 289]

**Function Parameters:**

- *string* **\$filename** The file path. See the class description to know what path to use.

**Computes the path of the file to load dynamically according to the user theme and the kind of file (kernel, module, menu or framework file).**

- **Access** protected

*string* function `Template::_include($parse_name)` [line 471]

**Function Parameters:**

- *string* **\$parse\_name** The identifier of the file to load. The two files must have been declared in the same template object.

**Include a file in another file. You can access to the same variables in the two variables, it's just as if you had copied the content of the included file in the includer.**

- **Deprecated**
- **Access** protected

*bool* function `Template::_load($parse_name)` [line 445]

**Function Parameters:**

- *string* **\$parse\_name** Path of the file to load (relative to the current file).

Loads a template file.

- **Access** protected

## Class Upload

*[line 40]*

This class provides you methods to upload easily files to the ftp.

- **Package** io
- **Author** Régis VIARRE <crowkait@phpboost.com

### Upload::\$base\_directory

*mixed = [line 227]*

### Upload::\$error

*mixed = " [line 43]*

### Upload::\$extension

*mixed = array() [line 228]*

### Upload::\$filename

*mixed = array() [line 229]*

Constructor *void* function Upload([*\$base\_directory* = 'upload']) *[line 51]*

#### **Function Parameters:**

- *string* ***\$base\_directory*** Set the directory to upload the files.

## constructor

*boolean* function Upload::file(\$filepostname, [\$regexp = "], [\$uniq\_name = false], [\$weight\_max = 100000000], [\$check\_exist = true]) [*line 65*]

### **Function Parameters:**

- *string* **\$filepostname** Name in the input file formulary.
- *string* **\$regexp** Regular expression to specify file format.
- *boolean* **\$uniq\_name** If true assign a new name if a file with same name already exist. Otherwise previous file is overwrite.
- *int* **\$weight\_max** The maximum file size.
- *boolean* **\$check\_exist** If true verify if a file with the same name exist on the ftp. Otherwise previous file is overwrite.

## Uploads a file.

*string* function Upload::validate\_img(\$filepath, \$width\_max, \$height\_max, [\$delete = true]) [*line 112*]

### **Function Parameters:**

- *string* **\$filepath** Path to the image.
- *int* **\$width\_max** Max width
- *int* **\$height\_max** Max height
- *boolean* **\$delete**

Checks whether an image is compliant to an maximum width and height, otherwise is **\$delete** value is true delete it.

## file.class.php

- **Package** io
- **Sub-Package** filesystem

ADD = true [line 31]  
CLOSEFILE = 0x1 [line 38]  
ERASE = false [line 30]  
LOCK = 0x4 [line 36]  
NOTCLOSEFILE = 0x2 [line 39]  
READ = 0x1 [line 33]  
READ\_WRITE = 0x3 [line 35]  
WRITE = 0x2 [line 34]

# file\_system\_element.class.php

- **Package** io
- **Sub-Package** filesystem

OPEN\_AFTER = false *[line 29]*  
OPEN\_NOW = true *[line 28]*

## Class File

*[line 47]*

This class represents a text file which can be read and written.

- **Package** io
- **Sub-Package** filesystem
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) > Nicolas Duhamel <akhenathon2@gmail.com>

### File::\$contents

*string = [line 345]*

- **Var** Content of the file

### File::\$fd

*File = [line 355]*

- **Var** descriptor of the open file.

## File::\$lines

*string[] = array()* [line 340]

- **Var** List of the lines of the file.

## File::\$mode

*int =* [line 350]

- **Var** Open mode

Constructor *void* function File::File(\$path, [\$mode = READ\_WRITE], [\$whenopen = OPEN\_AFTER]) [line 55]

### **Function Parameters:**

- *string* **\$path** Path of the file you want to work with.
- *int* **\$mode** If you want to open it only to read it, use the flag READ, if it's to write it use the WRITE flag, you also can use the READ\_WRITE flag.
- *bool* **\$whenopen** If you want to open the file now, use the OPEN\_NOW constant, if you want to open it only when you will need it, use the OPEN\_AFTER constant.

## **Builds a File object.**

*void* function File::close() [line 215]

**Closes a file and frees the allocated memory relative to the file.**

*void* function File::delete() [line 229]

**Deletes the file.**

*true* function File::finclude([\$once = true]) [line 294]

### **Function Parameters:**

- *bool* **\$once** true if you don't want to include it if it has already been included.

**Includes the file. Executes its PHP content here. Equivalent to the PHP include function.**

*void* function File::flush() [*line 280*]

**forces a write of all buffered output.**

*true* function File::frequire([\$once = true]) [*line 311*]

**Function Parameters:**

- *bool* **\$once** true if you don't want to include it if it has already been included.

**Requires the file. Executes its PHP content here. Equivalent to the PHP require function.**

*string* function File::get\_contents([\$start = 0], [\$len = -1]) [*line 112*]

**Function Parameters:**

- *int* **\$start** Byte from which you want to start. 0 if you want to read the file from its beginning, 1 to start with the second etc.
- *int* **\$len** Number of bytes you want to read.

**Returns the content of the file.**

*int* function File::get\_last\_access\_date() [*line 331*]

**Returns the last access date of the file.**

*int* function File::get\_last\_modification\_date() [*line 322*]

**Returns the date of the last modification of the file.**

*string[]* function File::get\_lines([\$start = 0], [\$n = -1], \$len) [*line 143*]

**Function Parameters:**

- *int* **\$start** Byte from which you want to start. 0 if you want to read the file from its beginning, 1 to start with the second etc.
- *int* **\$len** Number of bytes you want to read.
- **\$n**

**Returns the content of the file grouped by lines.**

*bool* function File::is\_open() [*line 245*]

**Allows you to know if the file is already open.**

*void* function File::lock([\$blocking = true]) [*line 254*]

**Function Parameters:**

- *bool* **\$blocking** if true, block the script, if false, non blocking operation

**Locks the file (it won't be readable by another thread which could try to access it).**

*void* function File::open() [*line 84*]

**Opens the file. You cannot read or write a closed file, use this method to open it.**

*void* function File::unlock() [*line 267*]

**Unlocks a file. The file must have been locked before you call this method.**

*bool* function File::write(\$data, [\$show = ERASE], [\$mode = CLOSEFILE], \$what) [*line 175*]

**Function Parameters:**

- *string* **\$data** The text you want to write in the file.
- *bool* **\$what** ERASE if you want to erase the file, ADD if you want to write at the end of the file.
- *bool* **\$mode** CLOSEFILE if you want to close the file before to write in it, NOTCLOSEFILE otherwise.
- **\$show**

**Writes some text in the file.**

## Class FileSystemElement

[*line 38*]

**This class represents any file system element.**

- **Package** io
- **Sub-Package** filesystem
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) > Nicolas Duhamel <akhenathon2@gmail.com>
- **Abstract Element**

### FileSystemElement::\$is\_open

*bool = [line 146]*

- **Var** Indicates whether the file system element is open or closed.

### FileSystemElement::\$path

*string = [line 141]*

- **Var** Path of the file system element

Constructor *void* function FileSystemElement::FileSystemElement(\$path) *[line 44]*

#### **Function Parameters:**

- *string* **\$path** Path of the element

### **Builds a FileSystemElement object from the path of the element.**

*void* function FileSystemElement::change\_chmod(\$chmod) *[line 123]*

#### **Function Parameters:**

- *int* **\$chmod** The new chmod of the file. Put a 0 at the beginning of the number to indicate to the PHP parser that it's an octal value.

### **Changes the chmod of the element.**

*void* function FileSystemElement::delete() [*line 135*]

**Deletes the element**

- **Abstract Element**

*bool* function FileSystemElement::exists() [*line 54*]

**Allows you to know if the file system element exists.**

*void* function FileSystemElement::get() [*line 75*]

**Initializes the file system element just before to be read.**

*string* function FileSystemElement::get\_name([\$full\_path = false], [\$no\_extension = false]) [*line 100*]

**Function Parameters:**

- *bool* **\$full\_path** True if you want the full path or false if you just want a relative path.
- *bool* **\$no\_extension** False if you want the name of the file with the extension and true without.

**Returns the element name.**

*void* function FileSystemElement::open() [*line 62*]

**Opens the file system element.**

*unknown\_type* function FileSystemElement::write() [*line 88*]

**Does the necessary treatment to apply at each writing operation**

- **Abstract Element**

# Class Folder

[line 37]

This class allows you to handle very easily a folder on the serveur.

- **Package** io
- **Sub-Package** filesystem
- **Author** Benoît Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) > Nicolas Duhamel <akhenathon2@gmail.com>

## Folder::\$files

*File[] = array()* [line 212]

- **Var** List of the files contained by this folder.

## Folder::\$folders

*Folder[] = array()* [line 217]

- **Var** List of the folders contained by this folder.

Constructor *void* function Folder::Folder(\$path, [\$whenopen = OPEN\_AFTER]) [line 44]

### **Function Parameters:**

- *string* **\$path** Path of the folder.
- *bool* **\$whenopen** OPEN\_AFTER if you want to synchronyse you with the folder only when it's necessary or OPEN\_NOW if you want to open it now.

## Builds a Folder object.

*True* function Folder::delete() [line 190]

**Deletes the folder and all what it contains.**

*FileSystemElement[]* function Folder::get\_all\_content() [*line 181*]

**Returns all the file system elements contained by the folder.**

*File[]* function Folder::get\_files([\$regex = ""]) [*line 103*]

**Function Parameters:**

- *string* **\$regex** PREG which describes the pattern the files you want to list must match. If you want all of them, don't use this parameter.

**Lists the files contained in this folder.**

*Folder* function Folder::get\_first\_folder() [*line 163*]

**Returns the first folder present in this folder**

*Folder[]* function Folder::get\_folders([\$regex = ""]) [*line 133*]

**Function Parameters:**

- *string* **\$regex** PREG which describes the pattern the folders you want to list must match. If you want all of them, don't use this parameter.

**Lists the folders contained in this folder.**

*void* function Folder::open() [*line 71*]

**Opens the folder.**



# Package members Procedural Elements

## authorizations.class.php

- **Package** members

AUTH\_CHILD\_PRIORITY = 0x02 [*line 29*]  
AUTH\_PARENT\_PRIORITY = 0x01 [*line 28*]

# groups.class.php

- **Package** members

```
ADMIN_NOAUTH_DEFAULT = false [line 28]  
GROUP_DEFAULT_IDSELECT = " [line 29]  
GROUP_DISABLED_ADVANCED_AUTH = true [line 31]  
GROUP_DISABLE_SELECT = 'disabled="disabled" ' [line 30]
```

# pm.class.php

- **Package members**

CHECK\_PM\_BOX = true *[line 29]*  
DEL\_PM\_CONVERS = true *[line 31]*  
NOCHECK\_PM\_BOX = false *[line 28]*  
SYSTEM\_PM = true *[line 30]*  
UPDATE\_MBR\_PM = true *[line 32]*

# session.class.php

- **Package** members

ALREADY\_HASHED = true *[line 32]*

AUTOCONNECT = true *[line 30]*

NO\_AUTOCONNECT = false *[line 31]*

SEASURF\_ATTACK\_ERROR\_PAGE = PATH\_TO\_ROOT.'/member/csrf-attack.php' *[line 33]*

# uploads.class.php

- **Package** members

ADMIN\_NO\_CHECK = true *[line 29]*  
EMPTY\_FOLDER = true *[line 28]*

# user.class.php

- **Package** members

GROUP\_TYPE = 2 *[line 29]*  
RANK\_TYPE = 1 *[line 28]*  
USER\_TYPE = 3 *[line 30]*

# Package members Classes

## Class Authorizations

[line 36]

This class contains only static methods, it shouldn't be instantiated.

- **Package** members
- **Author** Régis VIARRE < [crowkait@phpboost.com](mailto:crowkait@phpboost.com) > / Sautel Benoit <ben.popeye@phpboost.com>

Array function Authorizations::auth\_array\_simple(\$bit\_value, \$idselect, [\$admin\_auth\_default = true]) [line 85]

### Function Parameters:

- *int* **\$bit\_value** The bit emplacement in the authorization array.
- *string* **\$idselect** Html id of the html select field of authorizations (in most case the same value as \$bit\_value).
- *boolean* **\$admin\_auth\_default** Give authorization for the administrator by default.

Returns an array with the authorizations given by variable number of arrays passed in argument.

- **Static**

*array* function Authorizations::build\_auth\_array\_from\_form() [*line 45*]

Returns an array with the authorizations given by variable number of arrays passed in argument. This returned array is used to be serialized.

- **Static**

*array* function Authorizations::capture\_and\_shift\_bit\_auth(\$auth, \$original\_bit, [\$final\_bit = 1]) [*line 328*]

**Function Parameters:**

- *array* **\$auth** Array of authorizations.
- *int* **\$original\_bit** The bit to shift.
- *int* **\$final\_bit** Bit destination (1 is used by default).

Capture authorizations and shift a particular bit to an another bit (1 is used by default).

- **Static**

*boolean* function Authorizations::check\_auth(\$type, \$value, &\$array\_auth, \$bit, \$array\_auth) [*line 251*]

**Function Parameters:**

- *int* **\$type** Type of check, used RANK\_TYPE for ranks, GROUP\_TYPE for groups and USER\_TYPE for users.
- *int* **\$value** Value int the authorization array to check.
- *array* **\$array\_auth** Array of authorization.
- *int* **\$bit** Bit emplacement for the check
- **&\$array\_auth**

Check authorizations for a member, a group or a rank

- **Static**

String function Authorizations::generate\_select(\$auth\_bit, [\$array\_auth = array()], [\$array\_ranks\_default = array()], [\$idselect = "], [\$disabled = "], [\$disabled\_advanced\_auth = false]) [line 112]

**Function Parameters:**

- *int* **\$auth\_bit** The bit emplacement used to set it.
- *array* **\$array\_auth** Array of authorization, allow you to select value authorized for this bit.
- *array* **\$array\_ranks\_default** Array of ranks selected by default.
- *string* **\$idselect** Html id used for the select.
- *int* **\$disabled** Disabled all option for the select. Set to 1 for disable.
- *boolean* **\$disabled\_advanced\_auth** Disable advanced authorizations.

**Generate a multiple select field for the form which create authorization for ranks, groups and members.**

- **Static**

array function Authorizations::merge\_auth(\$parent, \$child, \$auth\_bit, \$mode) [line 287]

**Function Parameters:**

- *array* **\$parent** Array of authorizations.
- *array* **\$child** Array of authorizations.
- *int* **\$auth\_bit** Bit emplacement for the merge.
- *int* **\$mode** Mode used for the merge. Use AUTH\_PARENT\_PRIORITY to give to the parent the priority for the authorization, AUTH\_CHILD\_PRIORITY otherwise.

**Merge two authorizations array, first is the parent, second is the inherited child.**

- **Static**

# Class Group

[line 38]

This class provides methods to manage user in groups.

- **Package** members
- **Author** Régis VIARRE < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

## Group::\$groups\_auth

*mixed* = [line 137]

## Group::\$groups\_name

*mixed* = [line 136]

Constructor *void* function Group::Group(&\$groups\_info, \$groups\_info) [line 45]

### **Function Parameters:**

- *array* **\$groups\_info** Informations of all groups.
- **&\$groups\_info**

## Constructor. Loads informations groups.

*boolean* function Group::add\_member(\$user\_id, \$idgroup) [line 58]

### **Function Parameters:**

- *int* **\$user\_id** User id
- *int* **\$idgroup** Group id

## Adds a member in a group

*void* function Group::edit\_member(\$user\_id, \$array\_user\_groups) [line 84]

### **Function Parameters:**

- *int* **\$user\_id** The user id
- *array* **\$array\_user\_groups** The new array of groups.

**Edits the user groups, compute difference between previous and new groups.**

*array* function Group::get\_groups\_array() [*line 113*]

**Returns the array of user groups (id => name)**

*void* function Group::remove\_member(\$user\_id, \$idgroup) [*line 123*]

**Function Parameters:**

- *int* **\$user\_id** The user id
- *int* **\$idgroup** The id group.

**Removes a member in a group.**

## Class PrivateMsg

[*line 39*]

**This class provides methods to manage private message.**

- **Package** members
- **Author** Régis VIARRE < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

**PrivateMsg::\$pm\_convers\_id**

*mixed* = [*line 208*]

**PrivateMsg::\$pm\_msg\_id**

*mixed* = [*line 209*]

*int* function PrivateMsg::count\_conversations(\$userid) [*line 47*]

**Function Parameters:**

- *int* **\$userid** The user id.

## Counts the user's number of conversation.

*int* function PrivateMsg::delete(\$pm\_to, \$pm\_idmsg, \$pm\_idconvers) [*line 184*]

### **Function Parameters:**

- *int* **\$pm\_to**
- *int* **\$pm\_idmsg**
- *int* **\$pm\_idconvers**

## Deletes a private message, until the recipient has not read it.

*void* function PrivateMsg::delete\_conversation(\$pm\_userid, \$pm\_idconvers, \$pm\_expd, \$pm\_del, \$pm\_update) [*line 143*]

### **Function Parameters:**

- *int* **\$pm\_userid**
- *int* **\$pm\_idconvers**
- *int* **\$pm\_expd**
- *boolean* **\$pm\_del**
- *boolean* **\$pm\_update**

## Deletes a conversation.

*void* function PrivateMsg::send(\$pm\_to, \$pm\_idconvers, \$pm\_contents, \$pm\_from, \$pm\_status, [\$check\_pm\_before\_send = true]) [*line 109*]

### **Function Parameters:**

- *int* **\$pm\_to** The member's user id destination.
- *int* **\$pm\_idconvers**
- *string* **\$pm\_contents** The content of the answer.
- *int* **\$pm\_from** The member's user id author.
- *int* **\$pm\_status**
- *boolean* **\$check\_pm\_before\_send**

## Answers to a conversation

*void* function PrivateMsg::start\_conversation(\$pm\_to, \$pm\_objet, \$pm\_contents, \$pm\_from, [\$system\_pm = false])

[line 80]

#### **Function Parameters:**

- *int* **\$pm\_to** The member's user id destination.
- *string* **\$pm\_objet** The object of the conversation.
- *string* **\$pm\_contents** The content of the conversation.
- *int* **\$pm\_from** The member's user id author.
- *boolean* **\$system\_pm** If true, the conversation has been started by the system, and not by the private message interface.

**Starts a conversation with another member.**

## Class Session

[line 40]

**This class manages all sessions for the users.**

- **Package** members
- **Author** Régis VIARRE <crowkait@phpboost.com

#### **Session::\$autoconnect**

*mixed* = array() [line 45]

#### **Session::\$data**

*mixed* = array() [line 43]

#### **Session::\$session\_mod**

*mixed* = 0 [line 44]

*void* function Session::act() [line 51]

**Manage the actions for the session caused by the user (connection, disconnection).**

*void* function Session::check(\$session\_script\_title) [line 297]

#### **Function Parameters:**

- *string* **\$session\_script\_title** The page title where the session has been check.

## Check session validity, and update it

*true* function Session::csrf\_get\_protect([\$redirect = SEASURF\_ATTACK\_ERROR\_PAGE]) *[line 715]*

### **Function Parameters:**

- *mixed* **\$redirect** if string, redirect to the \$redirect error page if the token is wrong if false, do not redirect

## Check the session against CSRF attacks by GET. Checks that GETs are done from this site with a correct token.

*bool* function Session::csrf\_post\_protect([\$redirect = SEASURF\_ATTACK\_ERROR\_PAGE]) *[line 681]*

### **Function Parameters:**

- *mixed* **\$redirect** if string, redirect to the \$redirect error page if the token is wrong if false, do not redirect

## Check the session against CSRF attacks by POST. Checks that POSTs are done from this site. 2 different cases are accepted but the first is safer:

- The request contains a parameter whose name is token and value is the value of the token of the current session.
- If the token isn't in the request, we analyse the HTTP referer to be sure that the request comes from the current site and not from another which can be suspect  
If the request doesn't match any of these two cases, this method will consider that it's a CSRF attack.

*void* function Session::end() *[line 357]*

### **Destroy the session**

*void* function Session::garbage\_collector() *[line 546]*

### **Deletes all the existing sessions**

- **Static**

array function Session::get\_module\_parameters([\$module = "]) [*line 413*]

**Function Parameters:**

- *string* **\$module** module module name (if null then current module)

## Get module's parameters from session

string function Session::get\_token() [*line 657*]

**Return the session token**

void function Session::load() [*line 237*]

**Get informations from the user, and set it for his session.**

void function Session::set\_module\_parameters(\$parameters, [\$module = "]) [*line 379*]

**Function Parameters:**

- *mixed* **\$parameters** module's parameters
- **\$module**

## Save module's parameters into session

True function Session::start(\$user\_id, \$password, \$level, \$session\_script, \$session\_script\_get, \$session\_script\_title, [\$autoconnect = false], [\$already\_hashed = false]) [*line 139*]

**Function Parameters:**

- *int* **\$user\_id** The member's user id.
- *string* **\$password** The member's password.
- *string* **\$session\_script** Session script value where the session is started.
- *string* **\$session\_script\_get** Get value of session script where the session is started.
- *string* **\$session\_script\_title** Title of session script where the session is started.
- *boolean* **\$autoconnect** The member user id.
- *boolean* **\$already\_hashed** True if password has been already hashed width str\_hash() function, false otherwise.
- **\$level**

## Start the session

# Class Uploads

[line 36]

- **Package members**

## Uploads::\$base\_directory

*mixed = [line 369]*

## Uploads::\$error

*mixed = " [line 39]*

## Uploads::\$extension

*mixed = array() [line 370]*

## Uploads::\$filename

*mixed = array() [line 371]*

*void function Uploads::Add\_folder(\$id\_parent, \$user\_id, \$name) [line 44]*

### **Function Parameters:**

- **\$id\_parent**
- **\$user\_id**
- **\$name**

*void function Uploads::Del\_file(\$id\_file, \$user\_id, [\$admin = false]) [line 101]*

### **Function Parameters:**

- **\$id\_file**
- **\$user\_id**
- **\$admin**

*void function Uploads::Del\_folder(\$id\_folder) [line 75]*

### **Function Parameters:**

- **\$id\_folder**

*void function Uploads::Empty\_folder\_member(\$user\_id) [line 58]*

**Function Parameters:**

- **\$user\_id**

*void function Uploads::Find\_subfolder(\$array\_folders, \$id\_cat, &\$array\_child\_folder) [line 255]*

**Function Parameters:**

- **\$array\_folders**
- **\$id\_cat**
- **&\$array\_child\_folder**

*void function Uploads::get\_admin\_url(\$id\_folder, \$pwd, [\$member\_link = ""]) [line 270]*

**Function Parameters:**

- **\$id\_folder**
- **\$pwd**
- **\$member\_link**

*void function Uploads::get\_img\_mimetype(\$type) [line 308]*

**Function Parameters:**

- **\$type**

*void function Uploads::get\_url(\$id\_folder, \$pwd, \$popup) [line 285]*

**Function Parameters:**

- **\$id\_folder**
- **\$pwd**
- **\$popup**

*void function Uploads::Member\_memory\_used(\$user\_id) [line 300]*

**Function Parameters:**

- **\$user\_id**

*void function Uploads::Move\_file(\$move, \$to, \$user\_id, [\$admin = false]) [line 219]*

***Function Parameters:***

- **\$move**
- **\$to**
- **\$user\_id**
- **\$admin**

*void function Uploads::Move\_folder(\$move, \$to, \$user\_id, [\$admin = false]) [line 182]*

***Function Parameters:***

- **\$move**
- **\$to**
- **\$user\_id**
- **\$admin**

*void function Uploads::Rename\_file(\$id\_file, \$name, \$previous\_name, \$user\_id, [\$admin = false]) [line 155]*

***Function Parameters:***

- **\$id\_file**
- **\$name**
- **\$previous\_name**
- **\$user\_id**
- **\$admin**

*void function Uploads::Rename\_folder(\$id\_folder, \$name, \$previous\_name, \$user\_id, [\$admin = false]) [line 128]*

***Function Parameters:***

- **\$id\_folder**
- **\$name**
- **\$previous\_name**
- **\$user\_id**
- **\$admin**

# Class User

[line 37]

This class manage user, it provide you methods to get or modify user informations, moreover methods allow you to control user authorizations

- **Package** members
- **Author** Régis VIARRE < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

## User::\$groups\_auth

*mixed* = [line 271]

## User::\$user\_data

*mixed* = [line 270]

## User::\$user\_groups

*mixed* = [line 272]

Constructor *void* function User::User(\$session\_data, &\$groups\_info, \$groups\_info) [line 44]

### **Function Parameters:**

- *array* **\$session\_data**
- *array* **\$groups\_info**
- **&\$groups\_info**

**Sets global authorizations which are given by all the user groups authorizations.**

*boolean* function User::check\_auth(\$array\_auth\_groups, \$authorization\_bit) [line 121]

### **Function Parameters:**

- *array* **\$array\_auth\_groups** The array passed to check the authorization.
- *int* **\$authorization\_bit** Value of position bit to check the authorization. This value has to be a multiple of two. You can use this simplified scripture : 0x01, 0x02, 0x04, 0x08 to set a new position bit to check.

**Get the authorizations given by all the user groups. Then check the authorization.**

*boolean* function User::check\_level(\$secure) [*line 106*]

**Function Parameters:**

- *int* **\$secure** Constant of level authorization to check (MEMBER\_LEVEL, MODO\_LEVEL, ADMIN\_LEVEL).

## Check the authorization level

*unknown\_type* function User::check\_max\_value(\$key\_auth, [\$max\_value\_compare = 0]) [*line 141*]

**Function Parameters:**

- *int* **\$key\_auth**
- *int* **\$max\_value\_compare** Maximal value to compare

## Get the maximum value of authorization in all user groups.

*unknown\_type* function User::get\_attribute(\$attribute) [*line 66*]

**Function Parameters:**

- *string* **\$attribute** The attribute name.

## Accessor

*string* function User::get\_groups() [*line 164*]

### Get all user groups

*string* function User::get\_group\_color(\$user\_groups, [\$level = 0]) [*line 87*]

**Function Parameters:**

- *string* **\$user\_groups** The list of user groups separated by pipe.
- *int* **\$level** The user level. Only member have special color.

## Get the user group associated color.

- **Static**

*int* function User::get\_id() [*line 75*]

### **Get the user id**

*void* function User::set\_user\_lang(\$user\_lang) [*line 199*]

#### **Function Parameters:**

- *string* **\$user\_lang** The new lang

### **Modify the user lang.**

*void* function User::set\_user\_theme(\$user\_theme) [*line 173*]

#### **Function Parameters:**

- *string* **\$user\_theme** The new theme.

### **Modify the user theme.**

*void* function User::update\_user\_lang(\$user\_lang, \$user\_theme) [*line 208*]

#### **Function Parameters:**

- *string* **\$user\_theme** The new lang
- **\$user\_lang**

### **Modify the lang for guest in the database (sessions table).**

*void* function User::update\_user\_theme(\$user\_theme) [*line 182*]

#### **Function Parameters:**

- *string* **\$user\_theme** The new theme

**Modify the theme for guest in the database (sessions table).**



# Package menu Procedural Elements

## menu.class.php

- **Package** menu

```
BLOCK_POSITION__ALL = 9 [line 44]
BLOCK_POSITION__BOTTOM_CENTRAL = 4 [line 39]
BLOCK_POSITION__FOOTER = 6 [line 41]
BLOCK_POSITION__HEADER = 1 [line 36]
BLOCK_POSITION__LEFT = 7 [line 42]
BLOCK_POSITION__NOT_ENABLED = 0 [line 35]
BLOCK_POSITION__RIGHT = 8 [line 43]
BLOCK_POSITION__SUB_HEADER = 2 [line 37]
BLOCK_POSITION__TOP_CENTRAL = 3 [line 38]
BLOCK_POSITION__TOP_FOOTER = 5 [line 40]
MENU_AUTH_BIT = 1 [line 30]
MENU_ENABLED = true [line 32]
MENU_ENABLE_OR_NOT = 42 [line 31]
MENU_NOT_ENABLED = false [line 33]
MENU__CLASS = 'Menu' [line 28]
```

# Package menu Classes

## Class Menu

*[line 52]*

This class represents a menu element and is used to build any kind of menu

- **Package** menu
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >
- **Abstract Element**

### Menu::\$auth

*int[string] = null [line 205]*

- **Var** Represents the Menu authorisations array
- **Access** protected

### Menu::\$block

*int = BLOCK\_POSITION\_\_NOT\_ENABLED [line 215]*

- **Var** The Menu block position
- **Access** protected

## Menu::\$enabled

*bool = MENU\_NOT\_ENABLED [line 210]*

- **Var** true if the Menu is used
- **Access** protected

## Menu::\$id

*int = 0 [line 195]*

- **Var** the element identifier, only used by the service
- **Access** protected

## Menu::\$position

*int = -1 [line 220]*

- **Var** The Menu position on the website
- **Access** protected

## Menu::\$title

*string = " [line 200]*

- **Var** the Menu title
- **Access** protected

Constructor *void* function Menu::Menu(\$title, \$id) [line 60]

### **Function Parameters:**

- *string* **\$title** the Menu title

- *int* **\$id** its id in the database

## Build a Menu element.

*string* function Menu::admin\_display() [*line 130*]

### Display the menu admin gui

*string* function Menu::cache\_export() [*line 140*]

- **Abstract Element**

*string* function Menu::cache\_export\_begin() [*line 145*]

*string* function Menu::cache\_export\_end() [*line 155*]

*string* function Menu::display([*\$tpl = false*], *\$template*) [*line 120*]

#### **Function Parameters:**

- [Template](#) **\$template** the template to use
- **\$tpl**

## Display the menu

- **Abstract Element**

*void* function Menu::enabled([*\$enabled = MENU\_ENABLED*]) [*line 77*]

#### **Function Parameters:**

- *bool* **\$enabled** Enable or not the Menu

*array* function Menu::get\_auth() [*line 95*]

*int* function Menu::get\_block() [*line 103*]

*int* function Menu::get\_block\_position() [*line 107*]

*int* function Menu::get\_id() [*line 99*]

*string* function Menu::get\_title() [*line 91*]

*void* function Menu::id(\$id) [*line 165*]

**Function Parameters:**

- *int* **\$id** Set the Menu database id

*bool* function Menu::is\_enabled() [*line 111*]

*void* function Menu::set\_auth(\$auth, \$url) [*line 73*]

**Function Parameters:**

- *array* **\$url** the authorisation array to set
- **\$auth**

*int* function Menu::set\_block(\$block) [*line 81*]

**Function Parameters:**

- **\$block**

*void* function Menu::set\_block\_position(\$position) [*line 85*]

**Function Parameters:**

- *int* **\$position** the Menu position to set

*void* function Menu::set\_title(\$title, \$image) [*line 69*]

**Function Parameters:**

- *string* **\$image** the value to set
- **\$title**

*void* function Menu::\_assign(&\$template, \$template) [*line 174*]

**Function Parameters:**

- [Template](#) **\$template** the template on which we gonna assign vars
- **&\$template**

## Assign tpl vars

- **Access** protected

## content\_menu.class.php

- **Package** menu
- **Sub-Package** contentmenu

`CONTENT_MENU__CLASS = 'ContentMenu' [line 30]`

## Class ContentMenu

[line 38]

- **Package** menu
- **Sub-Package** contentmenu
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >

### ContentMenu::\$content

`string = " [line 97]`

- **Var** the menu's content

### ContentMenu::\$display\_title

`bool = true [line 102]`

- **Var** If true, the content menu title will be displayed

Constructor *void* function ContentMenu::ContentMenu(\$title) [line 41]

#### **Function Parameters:**

- **\$title**

*void* function ContentMenu::cache\_export() [*line 86*]

*a function* ContentMenu::display() [*line 74*]

**Display the content menu.**

*string* function ContentMenu::get\_content() [*line 68*]

*bool* function ContentMenu::get\_display\_title() [*line 63*]

**Returns true if the title will be displayed**

*void* function ContentMenu::set\_content(\$content) [*line 56*]

**Function Parameters:**

- *string* **\$content** the content to set

*void* function ContentMenu::set\_display\_title(\$display\_title) [*line 51*]

**Function Parameters:**

- *bool* **\$display\_title** if false, the title won't be displayed

# feed\_menu.class.php

- **Package** menu
- **Sub-Package** feedmenu

FEED\_MENU\_\_CLASS = 'FeedMenu' *[line 31]*

## Class FeedMenu

*[line 39]*

- **Package** menu
- **Sub-Package** feedmenu
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >

### FeedMenu::\$begin\_at

*mixed = 0 [line 134]*

### FeedMenu::\$category

*mixed = 0 [line 132]*

### FeedMenu::\$module\_id

*mixed = " [line 130]*

### FeedMenu::\$name

*mixed = " [line 131]*

### FeedMenu::\$number

*mixed = 10 [line 133]*

### FeedMenu::\$url

*string = " [line 129]*

- **Var** the feed url

Constructor *void* function FeedMenu::FeedMenu(\$title, \$module\_id, [\$category = 0], [\$name = DEFAULT\_FEED\_NAME], [\$number = 10], [\$begin\_at = 0]) *[line 42]*

**Function Parameters:**

- **\$title**
- **\$module\_id**
- **\$category**
- **\$name**
- **\$number**
- **\$begin\_at**

*void* function FeedMenu::cache\_export() *[line 95]*

*void* function FeedMenu::display() *[line 88]*

*string* function FeedMenu::get\_module\_id() *[line 56]*

*the function* FeedMenu::get\_template([\$name = "], [\$block\_position = BLOCK\_POSITION\_\_LEFT]) *[line 111]*

**Function Parameters:**

- *string* **\$name** The feed name
- *string* **\$block\_position** The identifier block position defined in the inherit class menu

## Returns the tpl to parse a feed

- **Static**

*Return function* FeedMenu::get\_url([\$relative = false]) *[line 62]*

**Function Parameters:**

- *bool* **\$relative** If false, compute the absolute url, else, returns the relative one

*void* function FeedMenu::set\_cat(\$value) *[line 82]*

**Function Parameters:**

- *int* **\$value** the feed's category

*void* function FeedMenu::set\_module\_id(\$value) [*line 78*]

***Function Parameters:***

- *string* **\$value** the feed's module\_id

*void* function FeedMenu::set\_name(\$value) [*line 86*]

***Function Parameters:***

- *string* **\$value** the feed's name

## links\_menu.class.php

- **Package** menu
- **Sub-Package** linksmenu

```
HORIZONTAL_MENU = 'horizontal' [line 34]
HORIZONTAL_SCROLLING_MENU = 'horizontal_scrolling' [line 37]
LINKS_MENU__CLASS = 'LinksMenu' [line 30]
TREE_MENU = 'tree' [line 35]
VERTICAL_MENU = 'vertical' [line 33]
VERTICAL_SCROLLING_MENU = 'vertical_scrolling' [line 36]
```

## links\_menu\_element.class.php

- **Package** menu
- **Sub-Package** linksmenu

```
LINKS_MENU_ELEMENT__CLASS = 'LinksMenuElement' [line 35]  
LINKS_MENU_ELEMENT__CLASSIC_DISPLAYING = false [line 37]  
LINKS_MENU_ELEMENT__FULL_DISPLAYING = true [line 36]
```

# links\_menu\_link.class.php

- **Package** menu
- **Sub-Package** linksmenu

LINKS\_MENU\_LINK\_\_CLASS = 'LinksMenuLink' [*line 30*]

## Class LinksMenu

[*line 47*]

Create a Menu with children. Children could be Menu or LinksMenuLink objects

- **Package** menu
- **Sub-Package** linksmenu
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >

### LinksMenu::\$elements

*LinksMenuElement[] = array()* [*line 249*]

- **Var** Direct menu children list
- **Access** protected

### LinksMenu::\$type

*string =* [*line 244*]

- **Var** menu's type
- **Access** protected

Constructor *void* function `LinksMenu::LinksMenu($title, $url, [$image = "], [$type = VERTICAL_SCROLLING_MENU], $id) [line 58]`

**Function Parameters:**

- *string* **\$title** Menu title
- *string* **\$url** Destination url
- *string* **\$image** Menu's image url relative to the website root or absolute
- *string* **\$type** Menu's type
- *int* **\$id** The Menu's id in the database

## Constructor

*void* function `LinksMenu::add($element) [line 81]`

**Function Parameters:**

- *LinksMenuElement* **\$element** the LinksMenuLink or Menu to add

## Add a single LinksMenuLink or (sub) Menu

*void* function `LinksMenu::add_array(&$menu_elements) [line 71]`

**Function Parameters:**

- *&LinksMenuElement[]* **&\$menu\_elements** A reference to a list of LinksMenuLink and / or Menu to add

## Add a list of LinksMenu or (sub)Menu to the current one

*string* function `LinksMenu::cache_export([$template = false]) [line 147]`

**Function Parameters:**

- **\$template**

*string* function `LinksMenu::display([$template = false], [$mode = LINKS_MENU_ELEMENT__CLASSIC_DISPLAYING]) [line 106]`

**Function Parameters:**

- [Template](#) **\$template** the template to use
- **\$mode**

## Display the menu

*LinksMenuElement[]* function LinksMenu::get\_children() [*line 209*]

*string[]* function LinksMenu::get\_menu\_types\_list() [*line 217*]

### static method which returns all the menu types

static method which returns all the menu types

- **Static**

*string* function LinksMenu::get\_type() [*line 197*]

*void* function LinksMenu::set\_type(\$type) [*line 204*]

#### Function Parameters:

- *string* **\$type** Type of the menu

## Sets the type of the menu

Sets the type of the menu

*void* function LinksMenu::update\_uid() [*line 94*]

### Update the menu uid

Update the menu uid

*void* function LinksMenu::\_parent(\$type) [*line 228*]

#### Function Parameters:

- *string* **\$type** the type of the menu

## Increase the Menu Depth and set the menu type to its parent one

- **Access** protected

## Class LinksMenuElement

[line 46]

A LinksMenuElement contains a Title, an url, and an image url

- **Package** menu
- **Sub-Package** linksmenu
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >
- **Abstract Element**

### LinksMenuElement::\$depth

*int = 0 [line 237]*

- **Var** Menu's depth
- **Access** protected

### LinksMenuElement::\$image

*string = " [line 227]*

- **Var** the image url. Could be relative to the website root or absolute
- **Access** protected

## LinksMenuElement::\$uid

*int* = null [*line 232*]

- **Var** Menu's uid
- **Access** protected

## LinksMenuElement::\$url

*string* = " [*line 222*]

- **Var** the LinksMenuElement url
- **Access** protected

Constructor *void* function LinksMenuElement::LinksMenuElement(\$title, \$url, [\$image = "], \$id) [*line 56*]

### **Function Parameters:**

- *int* **\$id** The Menu's id in the database
- *\$title* **\$title**
- *\$url* **\$url**
- *\$image* **\$image**

## Build a LinksMenuElement object

*string* function LinksMenuElement::cache\_export() [*line 166*]

**returns the string to write in the cache file**

*string* function LinksMenuElement::cache\_export\_begin() [*line 136*]

**returns the string to write in the cache file at the beginning of the Menu element**

*string* function LinksMenuElement::cache\_export\_end() [*line 145*]

**returns the string to write in the cache file at the end of the Menu element**

*string* function LinksMenuElement::display([\$template = false], [\$mode = LINKS\_MENU\_ELEMENT\_CLASSIC\_DISPLAYING]) [*line 158*]

### **Function Parameters:**

- **Template** **\$template** Template according to which the menu must be displayed. If it's not displayed, a default template will be used.
- **\$mode**

### Displays the menu according to the given template

Displays the menu according to the given template

- **Abstract Element**

*string* function LinksMenuElement::get\_image([\$compute\_relative\_url = true]) [*line 109*]

#### **Function Parameters:**

- *bool* **\$compute\_relative\_url** If true, computes relative urls to the website root

*int* function LinksMenuElement::get\_uid() [*line 85*]

#### **Returns the menu uid**

Returns the menu uid

*string* function LinksMenuElement::get\_url([\$compute\_relative\_url = true]) [*line 100*]

#### **Function Parameters:**

- *bool* **\$compute\_relative\_url** If true, computes relative urls to the website root

*void* function LinksMenuElement::set\_image(\$image) [*line 68*]

#### **Function Parameters:**

- *string* **\$image** the value to set

*void* function LinksMenuElement::set\_url(\$url) [*line 75*]

#### **Function Parameters:**

- *string* **\$url** the value to set

*void* function LinksMenuElement::update\_uid() [*line 92*]

### Update the menu uid

Update the menu uid

*void* function LinksMenuElement::\_assign(&\$template, [\$mode = LINKS\_MENU\_ELEMENT\_\_CLASSIC\_DISPLAYING], \$template) [*line 180*]

#### Function Parameters:

- *Template* **\$template** the template on which we gonna assign vars
- *int* **\$mode** in LINKS\_MENU\_ELEMENT\_\_CLASSIC\_DISPLAYING mode, the links menu is displayed. With the LINKS\_MENU\_ELEMENT\_\_FULL\_DISPLAYING mode, the authorization form is also shown.
- **&\$template**

### Assign tpl vars

- **Access** protected

*void* function LinksMenuElement::\_parent() [*line 211*]

### Increase the Menu Depth and set the menu type to its parent one

- **Access** protected

## Class LinksMenuLink

[*line 38*]

## A Simple menu link

- **Package** menu
- **Sub-Package** linksmenu
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com)>

Constructor *void* function LinksMenuLink::LinksMenuLink(\$title, \$url, [\$image = "], \$id) [*line 48*]

### **Function Parameters:**

- *string* **\$title** Menu title
- *string* **\$url** Destination url
- *string* **\$image** Menu's image url relative to the website root or absolute
- *int* **\$id** The Menu's id in the database

## Constructor

*string* function LinksMenuLink::cache\_export(\$template) [*line 76*]

### **Function Parameters:**

- *Template* **\$template** the template to use to display the link

*string* function LinksMenuLink::display(\$template, [\$mode = LINKS\_MENU\_ELEMENT\_\_CLASSIC\_DISPLAYING]) [*line 58*]

### **Function Parameters:**

- *Template* **\$template** the template to use
- **\$mode**

## Display the menu

# mini\_menu.class.php

- **Package** menu
- **Sub-Package** minimenu

MINI\_MENU\_\_CLASS = 'MiniMenu' [line 30]

## Class MiniMenu

[line 38]

- **Package** menu
- **Sub-Package** minimenu
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >

### MiniMenu::\$function\_name

*mixed* = " [line 55]

Constructor *void* function MiniMenu::MiniMenu(\$title, \$filename) [line 41]

#### **Function Parameters:**

- **\$title**
- **\$filename**

*string* function MiniMenu::cache\_export() [line 49]

## module\_mini\_menu.class.php

- **Package** menu
- **Sub-Package** modulesminimenu

MODULE\_MINI\_MENU\_\_CLASS = 'ModuleMiniMenu' [*line 30*]

## Class ModuleMiniMenu

[*line 38*]

- **Package** menu
- **Sub-Package** modulesminimenu
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >

### ModuleMiniMenu::\$filename

*mixed* = " [*line 66*]

Constructor *void* function ModuleMiniMenu::ModuleMiniMenu(\$module, \$filename, \$title) [*line 45*]

#### **Function Parameters:**

- *string* **\$title** its name (according the the module folder name)
- **\$module**
- **\$filename**

### Build a ModuleMiniMenu element.

*string* function ModuleMiniMenu::cache\_export() [*line 54*]

*void* function ModuleMiniMenu::get\_title() [*line 61*]



# Package modules Procedural Elements

## module\_interface.class.php

- **Package** modules

```
ACCES_DENIED = 2 [line 31]
FUNCTIONNALITY_NOT_IMPLEMENTED = 8 [line 33]
MODULE_ATTRIBUTE_DOES_NOT_EXIST = 16 [line 34]
MODULE_NOT_AVAILABLE = 1 [line 30]
```

```
module_interface. ----- begin : January 15, 2008
copyright : (C) 2008 Loïc Rouchon email : horn@phpboost.
```

```
#####
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

```
module_interface.class.php ----- begin : January
15, 2008 copyright : (C) 2008 Loïc Rouchon email : horn@phpboost.com
```

```
#####
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

```
#####
```

```
MODULE_NOT_YET_IMPLEMENTED = 4 [line 32]
```

# packages\_manager.class.php

- **Package** modules

```
CONFIG_CONFLICT = 3 [line 34]
DO_NOT_GENERATE_CACHE_AFTER_THE_OPERATION = false [line 40]
GENERATE_CACHE_AFTER_THE_OPERATION = true [line 39]
MODULE_ALREADY_INSTALLED = 2 [line 33]
MODULE_FILES_COULD_NOT_BE_DROPPED = 5 [line 36]
MODULE_INSTALLED = 0 [line 30]
MODULE_UNINSTALLED = 0 [line 31]
NOT_INSTALLED_MODULE = 4 [line 35]
PHP_VERSION_CONFLICT = 6 [line 37]
UNEXISTING_MODULE = 1 [line 32]
```

# Package modules Classes

## Class ModuleInterface

[line 44]

This Class allow you to call methods on a ModuleInterface extended class that you're not sure of the method's availability. It also provides a set of generic methods that you could use to integrate your module with others, or allow your module to share services.

- **Package** modules
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >

### ModuleInterface::\$attributes

*mixed*[string] = [line 282]

- **Var** the attributes dictionary
- **Access** protected

### ModuleInterface::\$enabled

*mixed* = false [line 284]

### ModuleInterface::\$errors

*int* = [line 277]

- **Var** error flag
- **Access** protected

### ModuleInterface::\$functionalities

*string[] = [line 272]*

- **Var** list of the functionalities provided
- **Access** protected

### ModuleInterface::\$id

*string = [line 257]*

- **Var** the module identifier
- **Access** protected

### ModuleInterface::\$infos

*mixed = [line 267]*

- **Var** module's informations contained in ini files
- **Access** protected

### ModuleInterface::\$name

*string = [line 262]*

- **Var** the module full name
- **Access** protected

Constructor *void* function ModuleInterface::ModuleInterface([\$moduleId = "], [\$error = 0]) *[line 52]*

**Function Parameters:**

- *string* **\$moduleId** the module id. It's the name of the folder in which the module is
- *int* **\$error** allow you to instantiate your module with an error code

## ModuleInterface constructor

*mixed* function ModuleInterface::functionality(\$functionality, [\$args = null]) *[line 198]*

**Function Parameters:**

- *string* **\$functionality** the name of the method you want to call
- *mixed* **\$args** the args you want to pass to the \$functionality method

**Check the existence of the functionality and if exists call it. If she's not available, the FUNCTIONNALITY\_NOT\_IMPLEMENTED flag is raised.**

*mixed* function ModuleInterface::get\_attribute(\$attribute) *[line 139]*

**Function Parameters:**

- *\$attribute* **\$attribute** the attribute identifier in the dictionary

*int* function ModuleInterface::get\_errors() *[line 186]*

*string* function ModuleInterface::get\_id() *[line 100]*

*mixed[]* function ModuleInterface::get\_infos() *[line 124]*

*string* function ModuleInterface::get\_name() *[line 116]*

*returns* function ModuleInterface::got\_error([\$error = 0]) *[line 175]*

**Function Parameters:**

- *int* **\$error** to check a specific error, 0 otherwise

**Returns the last error. If called with no arguments, returns true if an error has occurred otherwise, false. If the method got an argument,**

*bool* function ModuleInterface::has\_functionalities(\$functionalities) *[line 222]*

**Function Parameters:**

- *string[]* **\$functionalities** the names of the methods you want to check the availability

### Check the availability of the functionalities (hook)

*bool* function ModuleInterface::has\_functionality(\$functionality) [*line 212*]

**Function Parameters:**

- *string* **\$functionality** the name of the method you want to check the availability

### Check the availability of the functionality (hook)

*bool* function ModuleInterface::is\_enabled() [*line 108*]

*void* function ModuleInterface::set\_attribute(\$attribute, \$value) [*line 155*]

**Function Parameters:**

- *string* **\$attribute** the attribute identifier
- *mixed* **\$value** the value to set

### Set the \$value of the attribute identified by the string \$attribute.

*void* function ModuleInterface::set\_error([\$error = 0]) [*line 235*]

**Function Parameters:**

- *int* **\$error** the error flag to raised

### Set the flag error.

*void* function ModuleInterface::unset\_attribute(\$attribute) [*line 164*]

**Function Parameters:**

- *string* **\$attribute** the attribute identifier

### Delete the attribute and free its memory.

# Class ModulesDiscoveryService

[line 38]

This class is a **ModuleInterface** factory providing some services like mass operations (on several modules at the same time) or identifications methods to get all modules that provide a given functionality

- **Package** modules
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >

## ModulesDiscoveryService::\$available\_modules

*mixed* = [line 192]

## ModulesDiscoveryService::\$loaded\_modules

*mixed* = [line 191]

Constructor *void* function ModulesDiscoveryService::ModulesDiscoveryService() [line 43]

**Builds a new ModuleInterface factory**

*mixed*[string] function ModulesDiscoveryService::functionality(\$functionality, \$modules) [line 68]

**Function Parameters:**

- *string* **\$functionality** The method name to call on ModuleInterfaces
- *mixed*[string] **\$modules** The modules arguments in an array which keys are modules ids and values specifics arguments for those modules.

**Call the method call functionality on each specified modules**

*ModuleInterface*[] function ModulesDiscoveryService::get\_all\_modules() [line 92]

**Returns a list with all the modules in it, even with those that have no ModuleInterface.  
Useful to do generic operations on modules.**

*ModuleInterface*[] function ModulesDiscoveryService::get\_available\_modules([\$functionality = 'none'], [\$modulesList = array()]) [line 107]

### Function Parameters:

- *string* **\$functionality** the functionality name. By default, returns all availables modules interfaces.
- *ModuleInterface[]* **\$modulesList** If specified, only keep modules interfaces having the requested functionality. Else, search in all availables modules interfaces.

### Returns the ModuleInterface list.

*ModuleInterface* function ModulesDiscoveryService::get\_module([\$module\_id = ""]) [line 141]

### Function Parameters:

- *string* **\$module\_id** The module id.

### Returns the ModuleInterface of the module which id is \$module\_id.

## Class PackagesManager

[line 48]

This class enables you to manages the PHPBoost packages which are nothing else than the modules.

- **Package** modules
- **Author** Benoit Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

*int* function PackagesManager::install\_module(\$module\_identifier, [\$enable\_module = true], [\$generate\_cache = GENERATE\_CACHE\_AFTER\_THE\_OPERATION]) [line 65]

### Function Parameters:

- *string* **\$module\_identifier** Module identifier (name of its folder)
- *bool* **\$enable\_module** true if you want the module to be enabled, otherwise false.
- *bool* **\$generate\_cache** GENERATE\_CACHE\_AFTER\_THE\_OPERATION if you want to regenerate the cache after the installation or DO\_NOT\_GENERATE\_CACHE\_AFTER\_THE\_OPERATION if you want to go on a new

package operation.

## Installs a module.

- **Static**

*int* function PackagesManager::uninstall\_module(\$module\_id, \$drop\_files) [*line 191*]

### **Function Parameters:**

- *int* **\$module\_id** Module id (in the DB\_TABLE\_MODULES table)
- *bool* **\$drop\_files** true if you want the module files to be dropped, otherwise false.

## Uninstalls a module.

- **Static**



# Package util Procedural Elements

## captcha.class.php

- **Package util**

```
CAPTCHA_EASY = 1 [line 30]
CAPTCHA_HARD = 3 [line 32]
CAPTCHA_NORMAL = 2 [line 31]
CAPTCHA_VERY_EASY = 0 [line 29]
CAPTCHA_VERY_HARD = 4 [line 33]
```

# date.class.php

- **Package util**

```
DATE_FORMAT = 3 [line 36]
DATE_FORMAT_LONG = 4 [line 37]
DATE_FORMAT_SHORT = 2 [line 35]
DATE_FORMAT_TINY = 1 [line 34]
DATE_FROM_STRING = 4 [line 33]
DATE_NOW = 1 [line 30]
DATE_RFC822_F = 5 [line 38]
DATE_RFC822_FORMAT = 'D, d M Y H:i:s O' [line 41]
DATE_RFC3339_F = 6 [line 39]
DATE_RFC3339_FORMAT = 'Y-m-d\TH:i:s' [line 42]
DATE_TIMESTAMP = 0 [line 29]
DATE_YEAR_MONTH_DAY = 2 [line 31]
DATE_YEAR_MONTH_DAY_HOUR_MINUTE_SECOND = 3 [line 32]
TIMEZONE_AUTO = TIMEZONE_USER [line 44]
```

# images\_stats.class.php

- **Package util**

```
DRAW_LEGEND = true [line 31]
DRAW_VALUES = true [line 33]
FRANKLINBC_TTF = PATH_TO_ROOT.'/kernel/data/fonts/franklinbc.ttf' [line 35]
NO_ALLOCATE_COLOR = false [line 28]
NO_DRAW_LEGEND = false [line 30]
NO_DRAW_PERCENT = false [line 29]
NO_DRAW_VALUES = false [line 32]
```

# pagination.class.php

- **Package util**

LINK\_START\_PAGE = false *[line 29]*  
NO\_PREVIOUS\_NEXT\_LINKS = false *[line 28]*

## unusual\_functions.inc.php

- **Package util**

*string* function get\_server\_url\_page(\$path) [*line 93*]

**Function Parameters:**

- *string* **\$path** Path to add after the site path.

**Computes the path at which we must redirect the user when PHPBoost is not installed.**

*void* function securit\_register\_globals() [*line 38*]

**Unsets all the variables automatically set by the register\_globals option. This function must be called only if the register\_globals option is enable, otherwise it is useless.**

- **Author** PHPBB 3 < <http://www.phpbb.com/>>

# url.class.php

- **Package util**

```
SERVER_URL = $_SERVER['PHP_SELF'] [line 29]  
URL__CLASS = 'url' [line 28]
```

# Package util Classes

## Class Bench

*[line 32]*

This class is done to time a process easily. You choose when to start and when to stop.

- **Package** util
- **Author** loic

### Bench::\$duration

*int = 0 [line 84]*

- **Var** duration microtime
- **Access** protected

### Bench::\$start

*int = 0 [line 79]*

- **Var** start microtime
- **Access** protected

Constructor *void* function `Bench::Bench()` *[line 38]*

**starts the bench now**

*float* function `Bench::get_microtime()` *[line 68]*

**computes the time with a microsecond precision**

- **Static**
- **Access** protected

*void* function `Bench::stop()` *[line 45]*

**stops the bench now**

*string* function `Bench::to_string([$digits = 3])` *[line 55]*

**Function Parameters:**

- *int* **\$digits** the desired display precision

**returns the number formatted with \$digits floating numbers**

## Class `Captcha`

*[line 40]*

**This class provide you an easy way to prevent spam by bot in public formular.**

- **Package** util
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com) >

**`Captcha::$code`**

*mixed* = " *[line 412]*

### Captcha::\$difficulty

*mixed* = 2 [line 415]

### Captcha::\$font

*mixed* = './kernel/data/fonts/impact.ttf' [line 414]

### Captcha::\$gd\_loaded

*mixed* = false [line 410]

### Captcha::\$height

*mixed* = 50 [line 413]

### Captcha::\$instance

*mixed* = 0 [line 409]

### Captcha::\$width

*mixed* = 160 [line 411]

*void* function Captcha::Captcha() [line 45]

**Captcha constructor. It allows you to create multiple instance of captcha, and check if GD is loaded.**

*void* function Captcha::display() [line 202]

**Display the captcha image to the user, and set the code to decrypt in the database.**

*string* function Captcha::display\_form([\$Template = false]) [line 176]

#### **Function Parameters:**

- *object* **\$Template** (optional) The template used to create and display the captcha formular.

**Display captcha formular.**

*boolean* function Captcha::is\_available() [line 57]

**Checks if captcha is loaded, disabled for members.**

*boolean* function Captcha::is\_valid() [line 137]

**Check if the code is valid, then delete it in the database to avoid multiple attempts.**

*void* function Captcha::js\_require() [line 161]

**Javascript alert if the formular of the captcha code is empty.**

*void function Captcha::set\_difficulty(\$difficulty) [line 85]*

**Function Parameters:**

- *int* **\$difficulty** The difficulty : 0: Dictionary words, regular background, horizontal text. 1: blended word, regular background. 2: blended word + figures, irregular background. 3: blended word + figures, shadows. 4: blended word + figures, shadows.

**Modifies the level of difficulty to decrypt the code on the captcha image.**

*void function Captcha::set\_font(\$font) [line 128]*

**Function Parameters:**

- *string* **\$font** Font used for the text on the image.

**Modify font used for the text on the image.**

*void function Captcha::set\_height(\$height) [line 119]*

**Function Parameters:**

- *float* **\$height** Height of the image.

**Modify height of the image.**

*void function Captcha::set\_instance(\$instance) [line 101]*

**Function Parameters:**

- *int* **\$instance**

**Modify instance number.**

*void function Captcha::set\_width(\$width) [line 110]*

**Function Parameters:**

- *float* **\$width** Width of the image.

Modify width of the image.

*void* function `Captcha::update_instance()` [*line 69*]

**Updates the object instance.**

## Class Date

[*line 57*]

This class allows you to handle easily some dates. A date is a day and an hour (year, month, day, hour, minutes, seconds). It supports the most common formats and manages timezones. Here are the definitions of the 3 existing timezones:

- **System timezone:** it's the timezone of the server, configured by the hoster. For instance, if your server is in France, it should be GMT+1.
- **Site timezone:** it's the timezone of the central place of the site. For example, if your site deals with the italian soccer championship, it will be GMT+1.
- **User timezone :** each registered user can specify its timezone. It's particularly useful for people who visit some sites from a foreign country.

- **Package** util
- **Author** Benoit Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com)>

**Date::\$timestamp**

*int* = 0 [*line 395*]

- **Var** The timestamp of the current date

Constructor *void* function `Date::Date()` [*line 99*]

**Builds and initializes a date. It admits a variable number of parameters depending on the value of the first one. The second parameter allows us to chose what time referential we use to create the date:**

- **TIMEZONE\_SYSTEM** if that date comes from for example the database (dates must be stored under this referential).
- **TIMEZONE\_SITE** if it's an entry coming from the site (nearly never used).
- **TIMEZONE\_USER** if it's an entry coming from the user (it's own timezone will be used)

The first parameter determines how to initialize the date, here are the rules to use for the other parameters:

- **DATE\_NOW** will initialize the date to the current time.  
`$date = new Date(DATE_NOW);` will build a date with the current date.
- **DATE\_YEAR\_MONTH\_DAY** if you want to build a date from a specified day (year, month, day). In this case the following parameters are:
  - int The year (for instance 2009)
  - int The month (for example 11)
  - int The day (for example 09)

For example, `$date = new Date(DATE_YEAR_MONTH_DAY, TIMEZONE_USER, 2009, 11, 09);`

- **DATE\_YEAR\_MONTH\_DAY\_HOUR\_MINUTE\_SECOND** if you want to build a date from a specified time. Here is the rule for the following parameters:
  - int The year (for instance 2009)
  - int The month (for instance 11)
  - int The day (for instance 09)
  - int The hour (for instance 12)
  - int The minutes (for instance 34)
  - int The seconds (for instance 12)

For instance `$date = new Date(DATE_YEAR_MONTH_DAY_HOUR_MINUTE_SECOND, 2009, 11, 09, 12, 34, 12);`

- **DATE\_TIMESTAMP** which builds a date from a UNIX timestamp.  
 For example `$date = new Date(DATE_TIMESTAMP, time());` is equivalent to `$date = new Date(DATE_NOW);`
- **DATE\_FROM\_STRING** which decodes a date written in a string by matching a pattern you have to specify.  
 The pattern is easy to write: d for day, m for month and y for year.  
 For instance, if your third parameter is '24/12/2009' and the fourth is 'm/d/y', it will be the december 24th of 2009.  
 Here are the rules:

*bool function Date::check\_date(\$month, \$day, \$year) [line 352]*

*Function Parameters:*

- *int \$month* The month
- *int \$day* The day
- *int \$year* The year

Determines whether a date is correct. For example the february 31st is not correct.

- **Static**

*string* function Date::format([\$format = DATE\_FORMAT\_TINY], [\$referencial\_timezone = TIMEZONE\_USER]) [*line 228*]

**Function Parameters:**

- *int* **\$format** One of the following enumeration:
  - DATE\_FORMAT\_TINY for a tiny formatting (only month and day)
  - DATE\_FORMAT\_SHORT for a short formatting (month, day, year)
  - DATE\_FORMAT for a longer displaying (year, month, day, hour and minutes)
  - DATE\_FORMAT\_LONG for a total displaying (year, month, day, hour, minutes and seconds)
  - DATE\_RFC822\_F to format according to what the RFC822 announces
  - DATE\_RFC3339\_F to format according to what the RFC3339 announces
- *int* **\$referencial\_timezone** One of the following enumeration:
  - TIMEZONE\_SYSTEM
  - TIMZONE\_SITE
  - TIMEZONE\_USER

Formats the date to a particular format.

*int* function Date::get\_day() [*line 302*]

**Returns the day of the date**

*int* function Date::get\_hours() [*line 311*]

**Returns the hours of the date**

*int* function Date::get\_minutes() [*line 320*]

**Returns the minutes of the date**

*int* function Date::get\_month() [*line 293*]

**Returns the month of the date**

*int* function Date::get\_seconds() [*line 329*]

**Returns the seconds of the date**

*int* function Date::get\_timestamp() [*line 275*]

**Returns the timestamp associated to the date**

*int* function Date::get\_year() [*line 284*]

**Returns the year of the date**

*string* function Date::to\_date() [*line 338*]

**Exports the date according to the format YYYY-mm-dd**

## Class MiniCalendar

[*line 37*]

**This class enables you to retrieve easily a date entered by a user. If the user isn't in the same timezone as the server, the hour will be automatically recomputed.**

- **Package** util
- **Author** Benoit Sautel < [ben.popeye@phpboost.com](mailto:ben.popeye@phpboost.com) >

### MiniCalendar::\$date

*Date* = [*line 141*]

- **Var** The date it displays

### MiniCalendar::\$form\_name

*string* = " [*line 137*]

- **Var** The calendar name

#### MiniCalendar::\$num\_instance

*int* = 0 [line 129]

- **Var** The number of calendars created in that page (used to know if we have to load the javascript code)

#### MiniCalendar::\$style

*string* = " [line 133]

- **Var** The CSS properties of the calendar

Constructor *void* function MiniCalendar::MiniCalendar(\$form\_name) [line 44]

#### **Function Parameters:**

- *string* **\$form\_name** Name of the mini calendar in the HTML code (you will retrieve the data in that field). This name must be a HTML identifier.

**Builds a calendar which will be displayable.**

*string* function MiniCalendar::display() [line 86]

**Displays the mini calendar. You must call the display method in the same order as the calendars are displayed, because it requires a javascript code loading.**

*Date* function MiniCalendar::get\_date() [line 77]

**Returns the date**

*Date* function MiniCalendar::retrieve\_date(\$calendar\_name) [line 119]

#### **Function Parameters:**

- *string* **\$calendar\_name** Name of the calendar (HTML identifier).

Retrieves a date entered in a mini calendar.

- **Static**

*void* function MiniCalendar::set\_date(\$date) [*line 57*]

**Function Parameters:**

- *Date* **\$date** Date

Sets the date at which will be initialized the calendar.

*void* function MiniCalendar::set\_style(\$style) [*line 68*]

**Function Parameters:**

- *string* **\$style** The CSS properties

Sets the **CSS** properties of the element. You can use it if you want to customize the mini calendar, but the best solution is to redefine the template in your module. The template used is `framework/mini_calendar.tpl`.

## Class Pagination

[*line 38*]

This class enables you to manage easily a pagination system. It's very useful when you have a lot of items and you cannot display all of them. It also can generate the where clause to insert in your SQL query which selects the items.

- **Package** util
- **Author** Régis Viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

### Pagination::\$nbr\_end\_links

*mixed = 3 [line 192]*

### Pagination::\$nbr\_start\_links

*mixed = 3 [line 191]*

### Pagination::\$page

*mixed = [line 190]*

### Pagination::\$var\_page

*mixed = [line 193]*

Constructor *void* function `Pagination::Pagination()` [line 44]

### Buils a Pagination.

*string* function `Pagination::display($path, $total_msg, $var_page, $nbr_msg_page, $nbr_max_link, [$font_size = 11], [$previous_next = true], [$link_start_page = true])` [line 62]

#### **Function Parameters:**

- *string* **\$path** Adress with the `url()` function which abstracts the management of url rewriting in pagination links. You have to specify where the value of page will be placed in the adresse with %d (it will be replaced automatically) Example: `url('page.php?p=%d', 'page-%d.php')` //p is the variable name passed by the arguement `$var_page`.
- *string* **\$total\_msg** The total number of items.
- *string* **\$var\_page** The variable name used to get the page in the adress (in most case "p" is used).
- *string* **\$nbr\_msg\_page** The number of items per page.
- *string* **\$nbr\_max\_link** The maximum number of links displayed.
- *string* **\$font\_size** Links font size.
- *string* **\$previous\_next** Display links before and after pagination links, to go to the next/previous page.
- *string* **\$link\_start\_page** Underline link to the current page.

### Returns a list of links between pages.

*int* function `Pagination::get_current_page()` [line 145]

*int* function `Pagination::get_first_msg($nbr_msg_page, $var_page)` [line 135]

#### **Function Parameters:**

- *int* **\$nbr\_msg\_page** Number of messages per page.
- *string* **\$var\_page** The variable name used to get the page in the adress (in most case "p" is used).

Returns the first message of the current page displayed. It usually used in SQL queries. Example : `$Sql->query_while("SELECT n.contents FROM " . PREFIX . "news n " . $Sql->limit($Pagination->get_first_msg($CONFIG_NEWS['pagination_news'], $CONFIG_NEWS['pagination_news'], __LINE__, __FILE__));` For further informations, refer to the db package documentation.

## Class Stats

[line 43]

This class provides easy ways to create several type of charts.

- **Package** util
- **Author** regis viarre < [crowkait@phpboost.com](mailto:crowkait@phpboost.com)>

### Stats::\$array\_allocated\_color

*mixed* = array() [line 596]

### Stats::\$array\_color\_stats

*mixed* = array(array(224, 118, 27),array(48,149,53),array(254,249,52),array(102,133,237),array(204,42,38),array(53,144,189),array(102,102,153),array(236,230,208),array(213,171,1),array(182,0,51),array(193,73,0),array(25,119,128),array(182,181,177),array(102,133,237)) [line 592]

### Stats::\$color\_index

*mixed* = 0 [line 597]

### Stats::\$data\_stats

*mixed* = [line 594]

### Stats::\$decimal

*mixed = 1* [*line 598*]

#### Stats::\$nbr\_color

*mixed = 14* [*line 593*]

#### Stats::\$nbr\_entry

*mixed =* [*line 595*]

Constructor *void* function Stats::Stats() [*line 46*]

*Return* function Stats::draw\_ellipse(\$w\_arc, \$h\_arc, [\$img\_cache = ""], [\$height\_3d = 20], [\$draw\_percent = true], [\$draw\_legend = true], [\$font\_size = 10], [\$font = FRANKLINBC\_TTF]) [*line 97*]

#### **Function Parameters:**

- *int* **\$w\_arc** Ellipse width.
- *int* **\$h\_arc** Ellipse height.
- *string* **\$img\_cache** Name of the image store in cache.
- *int* **\$height\_3d** Height for the 3d ellipse effect.
- *boolean* **\$draw\_percent** Display percents around the ellipse.
- *boolean* **\$draw\_legend** Display a legend next to the ellipse.
- *string* **\$font\_size** Font size used for the legend.
- *string* **\$font** Font type used for the legend.

### Drawn an ellipse.

*void* function Stats::draw\_graph() [*line 453*]

#### **Draw graph, not yet implemented.**

*Return* function Stats::draw\_histogram(\$w\_histo, \$h\_histo, [\$img\_cache = ""], [\$scale\_legend = array()], [\$draw\_legend = true], [\$draw\_values = true], [\$font\_size = 10], [\$font = FRANKLINBC\_TTF]) [*line 251*]

#### **Function Parameters:**

- *int* **\$w\_histo** Histogram width.
- *int* **\$h\_histo** Histogram height.
- *string* **\$img\_cache** Name of the image store in cache.
- *int* **\$scale\_legend** Height for the 3d histogram effect.
- *boolean* **\$draw\_legend** Display a legend next the histogram
- *boolean* **\$draw\_values** Display value under the histogram
- *string* **\$font\_size** Font size used for the legend.
- *string* **\$font** Font type used for the legend.

### Draw a histogram

*void* function Stats::load\_data(\$array\_stats, [\$draw\_type = 'ellipse'], [\$decimal = 1]) [*line 56*]

### Function Parameters:

- *array* **\$array\_stats** Associative array, array key is the legend for the value.
- *string* **\$draw\_type** Type of chart.
- *int* **\$decimal** Round precision.

Load data for the charts.

## Class Url

[line 44]

This class offers a simple way to transform an absolute or relative link to a relative one to the website root. It can also deals with absolute url and will convert only those from this site into relatives ones. Usage :

- In content, get the url with the `absolute()` method. It will allow content include at multiple level
- In forms, get the url with the `relative()` method. It's a faster way to display url

- **Package** util
- **Author** Loïc Rouchon < [horn@phpboost.com](mailto:horn@phpboost.com) >

### Url::\$is\_relative

*mixed* = false [line 526]

### Url::\$path\_to\_root

*mixed* = " [line 527]

### Url::\$server\_url

*mixed* = " [line 528]

### Url::\$url

*mixed* = " [line 525]

Constructor *void* function `Url::Url([$url = '.'], [$path_to_root = null], [$server_url = null])` [line 54]

### Function Parameters:

- *string* **\$url** the url string relative to the current path, to the website root if beginning with a "/" or an absolute url
- *string* **\$path\_to\_root** url context. default is PATH\_TO\_ROOT
- **\$server\_url**

**Build a Url object. By default, builds an Url object representing the current path. If the url is empty, no computation is done and an empty string will be returned when asking for both relative and absolute form of the url.**

*string* function Url::absolute() [*line 154*]

### Returns the absolute url

*true* function Url::check\_wellformness(\$url, [\$protocol = REGEX\_MULTIPLICITY\_OPTIONNAL], [\$user = REGEX\_MULTIPLICITY\_OPTIONNAL], [\$domain = REGEX\_MULTIPLICITY\_OPTIONNAL], [\$folders = REGEX\_MULTIPLICITY\_OPTIONNAL], [\$file = REGEX\_MULTIPLICITY\_OPTIONNAL], [\$args = REGEX\_MULTIPLICITY\_OPTIONNAL], [\$anchor = REGEX\_MULTIPLICITY\_OPTIONNAL], [\$forbid\_js = true]) [*line 516*]

### Function Parameters:

- *int* **\$protocol** REGEX\_MULTIPLICITY\_OPTION for the protocol sub-regex
- *int* **\$user** REGEX\_MULTIPLICITY\_OPTION for the user:password@ sub-regex
- *int* **\$domain** REGEX\_MULTIPLICITY\_OPTION for the domain sub-regex
- *int* **\$folders** REGEX\_MULTIPLICITY\_OPTION for the folders sub-regex
- *int* **\$file** REGEX\_MULTIPLICITY\_OPTION for the file sub-regex
- *int* **\$args** REGEX\_MULTIPLICITY\_OPTION for the arguments sub-regex
- *int* **\$anchor** REGEX\_MULTIPLICITY\_OPTION for the anchor sub-regex
- *bool* **\$forbid\_js** true if you want to forbid javascript uses in urls
- **\$url**

**Returns true if the url match the requested url form**

- **See** REGEX\_MULTIPLICITY\_ALL
- **See** REGEX\_MULTIPLICITY\_NOT\_USED
- **See** REGEX\_MULTIPLICITY\_AT\_LEAST\_ONE
- **See** REGEX\_MULTIPLICITY\_NEEDED
- **See** REGEX\_MULTIPLICITY\_OPTIONNAL
- **Static**

*string* function `Url::compress($url)` [*line 173*]

**Function Parameters:**

- *string* **\$url** the url to compress

## Compress a url by removing all "folder/.." occurrences

- **Static**

*string* function `Url::get_absolute_root()` [*line 217*]

## Returns the absolute website root Url

- **Static**

*string* function `Url::get_relative($url, [$path_to_root = null], [$server_url = null])` [*line 402*]

**Function Parameters:**

- *string* **\$url** the url to "relativize"
- *string* **\$path\_to\_root** Path to root of the page to which you want to fit the URL
- *string* **\$server\_url** Path from the site root of the page to which you want to fit the URL.

- **Static**

*the* function `Url::get_wellformness_regex([$protocol = REGEX_MULTIPLICITY_OPTIONNAL], [$user = REGEX_MULTIPLICITY_OPTIONNAL], [$domain = REGEX_MULTIPLICITY_OPTIONNAL], [$folders = REGEX_MULTIPLICITY_OPTIONNAL], [$file = REGEX_MULTIPLICITY_OPTIONNAL], [$args = REGEX_MULTIPLICITY_OPTIONNAL], [$anchor = REGEX_MULTIPLICITY_OPTIONNAL], [$forbid_js = true])` [*line 460*]

**Function Parameters:**

- *int* **\$protocol** REGEX\_MULTIPPLICITY\_OPTION for the protocol sub-regex
- *int* **\$user** REGEX\_MULTIPPLICITY\_OPTION for the user:password@ sub-regex
- *int* **\$domain** REGEX\_MULTIPPLICITY\_OPTION for the domain sub-regex
- *int* **\$folders** REGEX\_MULTIPPLICITY\_OPTION for the folders sub-regex
- *int* **\$file** REGEX\_MULTIPPLICITY\_OPTION for the file sub-regex
- *int* **\$args** REGEX\_MULTIPPLICITY\_OPTION for the arguments sub-regex
- *int* **\$anchor** REGEX\_MULTIPPLICITY\_OPTION for the anchor sub-regex
- *bool* **\$forbid\_js** true if you want to forbid javascript uses in urls

## Returns the regex matching the requested url form

- **See** REGEX\_MULTIPPLICITY\_ALL
- **See** REGEX\_MULTIPPLICITY\_NOT\_USED
- **See** REGEX\_MULTIPPLICITY\_AT\_LEAST\_ONE
- **See** REGEX\_MULTIPPLICITY\_NEEDED
- **See** REGEX\_MULTIPPLICITY\_OPTIONNAL
- **Static**

*string* function Url::html\_convert\_absolute2root\_relative(\$html\_text, [\$path\_to\_root = PATH\_TO\_ROOT], [\$server\_url = SERVER\_URL]) [*line 257*]

### Function Parameters:

- *string* **\$html\_text** The HTML text in which we gonna search for absolutes urls to convert into relatives ones.
- *string* **\$path\_to\_root** Path to root of the page to which you want to fit the URL.
- *string* **\$server\_url** Path from the site root of the page to which you want to fit the URL.

## Returns the HTML text with only relatives urls

- **Static**

*string* function Url::html\_convert\_root\_relative2absolute(\$html\_text, [\$path\_to\_root = PATH\_TO\_ROOT], [\$server\_url = SERVER\_URL]) [*line 232*]

### Function Parameters:

- *string* **\$html\_text** The HTML text in which we gonna search for root relatives urls (only those beginning by '/') to convert into absolutes ones.
- *string* **\$path\_to\_root** Path to root of the page to which you want to fit the URL.
- *string* **\$server\_url** Path from the site root of the page to which you want to fit the URL.

### Returns the HTML text with only absolutes urls

- **Static**

*string* function Url::html\_convert\_root\_relative2relative(\$html\_text, [\$path\_to\_root = PATH\_TO\_ROOT], [\$server\_url = SERVER\_URL]) [*line 282*]

#### **Function Parameters:**

- *string* **\$html\_text** The HTML text in which you want to replace the paths
- *string* **\$path\_to\_root** Path to root of the page to which you want to fit the URL.
- *string* **\$server\_url** Path from the site root of the page to which you want to fit the URL.

### Transforms the relative URL whose base is the site root (for instance /images/mypic.png) to the real relative path fitted to the current page.

- **Static**

*bool* function Url::is\_relative() [*line 129*]

*string* function Url::path\_to\_root([\$path = null]) [*line 415*]

#### **Function Parameters:**

- *string* **\$path** the new PATH\_TO\_ROOT to use

**Overrides the used PATH\_TO\_ROOT. if the argument is null, the value is only returned. Please note this is a PHP4 hack to allow a Class variable.**

- **Static**

*string* function `Url::relative()` [*line 138*]

**Returns the relative url if defined, else the absolute one**

*string* function `Url::root_to_local()` [*line 197*]

**Returns the relative path from the website root to the current path if working on a relative url**

*string* function `Url::server_url([$url = null], $path)` [*line 432*]

**Function Parameters:**

- *string* **\$path** the new SERVER URL to use
- **\$url**

**Overrides the used SERVER URL. if the argument is null, the value is only returned. Please note this is a PHP4 hack to allow a Class variable.**

- **Static**



# Package phpboost Procedural Elements

## functions.inc.php

- **Package** phpboost

ADDSLASHES\_AUTO = 0 [line 32]

ADDSLASHES\_FORCE = 1 [line 34]

ADDSLASHES\_NONE = 2 [line 36]

void function array\_combine(\$keys, \$values) [line 1107]

**Function Parameters:**

- **\$keys**
- **\$values**

bool function check\_mail(\$mail) [line 637]

**Function Parameters:**

- *string* **\$mail** String to check

**Checks if a string could match an email address form.**

- **See** Mail::check\_validity
- **Deprecated**

bool function check\_nbr\_links(\$contents, \$max\_nbr) [line 614]

**Function Parameters:**

- *string* **\$contents** String in which you want to count the number of links
- *int* **\$max\_nbr** Maximum number of links accepted.

**Checks if a string contains less than a defined number of links (used to prevent SPAM).**

`CLASS_IMPORT = '.class.php' [line 1190]`

`bool function delete_file($file) [line 943]`

**Function Parameters:**

- *string* **\$file** Path of the file to delete

**Deletes a file**

- **Deprecated**

`The function display_comments($script, $idprov, $vars, [$module_folder = "]) [line 313]`

**Function Parameters:**

- *string* **\$script**
- *int* **\$idprov** The data base id of the item for which you want to display the commenting interface.
- *string* **\$vars** The URL of the current page (the comments API will always redirect the user to the current page). You just have to add a 'com' HTTP parameter for which the value must be %s (it will be used by the comments API).
- *string* **\$module\_folder** The identifier of your module (the name of its folder).

**Returns the HTML code of the comments manager.**

`The function display_editor([$field = 'contents'], [$forbidden_tags = array()]) [line 291]`

**Function Parameters:**

- *string* **\$field** The name of the HTTP parameter which you will retrieve the value entered by the user.
- *string[]* **\$forbidden\_tags** The list of the tags you don't want to appear in the editor.

Returns the HTML code of the user editor. It uses the `ContentFormattingFactory` class, it allows you to write less code lines.

*string* function `file_get_contents_emulate($filename, [$inpath = false], [$resource_context = null])` *[line 1016]*

**Function Parameters:**

- *string* **\$filename** File to read.
- *string* **\$inpath** **\$inpath** See the PHP documentation
- *string* **\$resource\_context** **\$resource\_context** See the PHP documentation

**Emulates the PHP `file_get_contents_emulate`.**

*string* function `find_require_dir($dir_path, $require_dir, [$fatal_error = true])` *[line 506]*

**Function Parameters:**

- *string* **\$dir\_path** Path of the folder in which you want to search
- *string* **\$require\_dir** Default folder
- *string* **\$fatal\_error** true if you want to throw a fatal error if no file could be found, false otherwise.

Finds a folder according to the user language. You find the file in a folder in which there is one folder per lang. If it doesn't exist, you want to choose the file in another language. This function returns the path of an existing file (if the required lang exists, it will be it, otherwise it will be one of the existing files).

*string* function `get_ini_config($dir_path, $require_dir, [$ini_name = 'config.ini'])` *[line 472]*

**Function Parameters:**

- *string* **\$dir\_path** Path in which is the file (stop just before the lang folder)
- *string* **\$require\_dir** Lang folder in which must be the file
- *string* **\$ini\_name** Config file name

Returns the config field of a module configuration file. In fact, this field contains the default module configuration in which we can find some " characters. To solve the problem, this field is considered as a comment and when we want to retrieve its value, we have to call this method which returns its value.

*string* function `get_module_name()` *[line 543]*

**Retrieves the identifier (name of the folder) of the module which is currently executed.**

The function `get_start_page()` [line 600]

**Retrieves the site start page.**

`int` function `get_uid()` [line 1184]

**Returns a unique identifier (useful for example to generate some javascript ids)**

`string` function `get_ulang()` [line 242]

**Returns the current user's language.**

`string` function `get_utheme()` [line 232]

**Returns the current user's theme.**

`string` function `gmdate_format($format, [$timestamp = false], [$timezone_system = 0])` [line 786]

**Function Parameters:**

- **`string $format`** Formatting name (`date_format`, `date_format_tiny`, `date_format_short`, `date_format_long`)
- **`int $timestamp`** Time to format (UNIX timestamp)
- **`int $timezone_system`** Time zone (1 for the site time zone, 2 for the server time zone and 0 for the user timezone)

**Formats a date according to a specific form.**

- See [Date::format\(\)](#)
- **Deprecated**

`void` function `htmlspecialchars_decode($string, [$quote_style = null])` [line 1070]

**Function Parameters:**

- **`$string`**
- **`$quote_style`**

`void` function `html_entity_decode($string, [$quote_style = ENT_COMPAT], [$charset = null])` [line 1044]

**Function Parameters:**

- **\$string**
- **\$quote\_style**
- **\$charset**

HTML\_NO\_PROTECT = false *[line 29]*

HTML\_PROTECT = true *[line 30]*

void function import(\$path, [\$import\_type = CLASS\_IMPORT]) *[line 1201]*

**Function Parameters:**

- *string* **\$path** Path of the file to load without .class.php or .inc.php extension (for instance util/date)
- *string* **\$import\_type** the import type. Default is CLASS\_IMPORT, but you could also import a library by using LIB\_IMPORT (file whose extension is .inc.php) or INC\_IMPORT to include a .inc.php file (for example the current file, functions.inc.php).

## Umports a class or a lib from the framework

bool function inc(\$file, [\$once = true]) *[line 1236]*

**Function Parameters:**

- *string* **\$file** the file to include with an absolute path from the website root
- *bool* **\$once** if false use include instead of include\_once

## Includes a file

INC\_IMPORT = '.inc.php' *[line 1191]*

LIB\_IMPORT = '.lib.php' *[line 1192]*

string[] function load\_ini\_file(\$dir\_path, \$require\_dir, [\$ini\_name = 'config.ini']) *[line 383]*

**Function Parameters:**

- *string* **\$dir\_path** Path of the file (relative from this page).
- *string* **\$require\_dir** The name of the folder in which the configuration file should be. This folder must be in the bases file (\$dir\_path). If this directory doesn't exist, another will be read.
- *string* **\$ini\_name** The name of the configuration file you want to know.

**Loads a configuration file. You choose a bases path, and you specify a folder name in which you file should be found, if it doesn't exist, it will take a file in another folder. It's very interesting when you want to**

void function load\_menu\_lang(\$menu\_name) *[line 370]*

### Function Parameters:

- *string* **\$menu\_name** The identifier of the menu for which you want to load the lang file.

**Loads a menu lang file. It will load alone the file corresponding to the user lang, but if it doesn't exist, another lang will be choosen. An error will be displayed on the page and the script execution will be stopped if no lang file is found for this menu.**

*void* function load\_module\_lang(\$module\_name, [\$path = PATH\_TO\_ROOT]) [*line 327*]

### Function Parameters:

- *string* **\$module\_name** The identifier of the module for which you want to load the lang file.
- *string* **\$path** Path of the folder in which is the file. This path mustn't finish by the / character.

**Loads a module lang file. It will load alone the file corresponding to the user lang, but if it doesn't exist, another lang will be choosen. An error will be displayed on the page and the script execution will be stopped if no lang file is found for this module.**

MAGIC\_QUOTES\_DISABLED = false [*line 37*]

NO\_EDITOR\_UNPARSE = false [*line 40*]

NO\_FATAL\_ERROR = false [*line 39*]

NO\_UPDATE\_PAGES = true [*line 38*]

*string* function number\_round(\$number, \$dec) [*line 1004*]

### Function Parameters:

- *mixed* **\$number** Number to round
- *int* **\$dec** The number of decilam points

## Rounds a number

*mixed* function numeric(\$var, [\$type = 'int']) [*line 209*]

### Function Parameters:

- *string* **\$var** The value you want to convert.
- *string* **\$type** 'int' if you want to convert to an integer value, 'float' if you want a floating value.

**Converts a string to a numeric value.**

*bool* function of\_class(&\$object, \$classname, \$object) [line 1261]

**Function Parameters:**

- *object* **\$object** the object to check its type
- *string* **\$classname** the classname you want to compare with
- **&\$object**

### Tells if an object is an instance of a class

*int[]* function pages\_displayed([\$no\_update = false]) [line 965]

**Function Parameters:**

- *bool* **\$no\_update** True if you just want to read the number of pages viewed, false if you want to increment it.

**This function is called by the kernel on each displayed page to count the number of pages seen at each hour.**

*string[]* function parse\_ini\_array(\$links\_format) [line 409]

**Function Parameters:**

- *string* **\$links\_format** Serialized array

**Parses a table written in a special syntax which is user-friendly and can be inserted in a ini file (PHP serialized arrays cannot be inserted because they contain the " character). The syntax is very easy, it really looks like the PHP array declaration: key => value, key2 => value2 You can nest some elements: key => (key1 => value1, key2 => value2), key2 => value2**

*string* function phpboost\_version() [line 1322]

**Returns the full phpboost version with its build number**

*void* function redirect(\$url) [line 556]

**Function Parameters:**

- *string* **\$url** URL at which you want to redirect the user.

Redirects the user to the URL and stops purely the script execution (database deconnexion...).

*void function redirect\_confirm(\$url\_error, \$\_error, [\$delay\_redirect = 3]) [line 580]*

**Function Parameters:**

- *string* **\$url\_error** Url at which you want to redirect him.
- *string* **\$\_error** The message to show him
- *int* **\$delay\_redirect** Number of seconds after which you want him to be redirected.

Displays a confirmation message during a defined delay and then redirects the user.

*void function req(\$file, [\$once = true]) [line 1211]*

**Function Parameters:**

- *string* **\$file** the file to require with an absolute path from the website root
- *bool* **\$once** if false use require instead of require\_once

Requires a file

*mixed function retrieve(\$var\_type, \$var\_name, \$default\_value, [\$force\_type = NULL], [\$flags = 0]) [line 72]*

**Function Parameters:**

- *int* **\$var\_type** The origin of the variable: GET if it's a parameter in the request URL, POST if the variable was in a formulary, COOKIE if the variables come from a cookie and FILES if it's a file.
- *string* **\$var\_name** Name of a HTTP variable you want to retrieve.
- *mixed* **\$default\_value** The value you want the variable you retrieve has if the HTTP parameter doesn't exist.
- *string* **\$force\_type** Type of the variable you want to retrieve. If you don't use this parameter, the returned variable will have the same type as the default value you imposed. When you force the variable type, a cast operation will be made from string (it's a string in the HTTP request) to the type you choosed. The types you can use are numerous:
  - TINTEGER to retrieve an integer value.
  - TSTRING to retrieve a string. The HTML code in this string is protected (XSS protection) and the dangerous MySQL characters are escaped. You can use this variable directly in a MySQL query.  
It you want to use it now without inserting it in a data base, use the stripslashes PHP function.
  - TSTRING\_UNCHANGE if you want to retrieve the value of a string without any processing (no quotes escaping and no HTML protection).
  - TSTRING\_PARSE if you want to parse the value you retrieved. The HTML code is

protected, it parses with the user parser and the quotes are escaped. Ready to be inserted in a MySQL query !

- TBOOL to retrieve a boolean value.
  - TUNSIGNED\_INT if you expect an unsigned integer.
  - TUNSIGNED\_DOUBLE to retrieve an unsigned double value.
  - TSTRING\_HTML if you don't want to protect the HTML code of the content but you want to escape the quotes.
  - TSTRING\_AS\_RECEIVED if you want to retrieve the string variable as it was in the HTTP request. Be careful, if the magic\_quotes are enabled (use the MAGIC\_QUOTES constant to know it), the quotes are escaped, otherwise they aren't.
  - TARRAY to retrieve an array. The values it contains aren't processed.
  - TDOUBLE to retrieve a double value
  - TNONE if you want to get the input variable as it has been received (the return value will be a string because HTTP parameters are all strings).
- **int \$flags** You can change the behaviour of this method: USE\_DEFAULT\_IF\_EMPTY will allow you to retrieve the default value even if the parameter exists but its value is empty (to know if the var is empty, we use the empty() PHP function).

**Retrieves an input variable. You can retrieve any parameter of the HTTP request which launched the execution of this page.**

*string* function second\_parse(&\$content, \$content, \$forbidden\_tags, \$addslashes) [line 699]

**Function Parameters:**

- *string* **\$content** Content to second parse
- *string[]* **\$forbidden\_tags** List of the forbidden formatting tags
- *bool* **\$addslashes** if true, the second parsed string will be escaped.
- **&\$content**

**Second parses a string with several default parameters. This methods exists to lighten the number of lines written.**

- See [ContentSecondParser](#)

*string* function second\_parse\_url(&\$url, \$url) [line 716]

**Function Parameters:**

- *string* **\$url** Url to second parse
- **&\$url**

## Second parses relative urls to absolute urls.

- See [Url](#)

*string* function `set_subregex_multiplicity($sub_regex, $multiplicity_option, $occurence)` [line 1295]

### **Function Parameters:**

- *string* **\$sub\_regex** the sub-regex on which add the multiplicity
- *int* **\$occurence** REGEX\_MULTIPPLICITY\_OPTION
- **\$multiplicity\_option**

## Returns the sub-regex with its multiplicity option

- **See** REGEX\_MULTIPPLICITY\_NOT\_USED
- **See** REGEX\_MULTIPPLICITY\_ALL
- **See** REGEX\_MULTIPPLICITY\_AT\_LEAST\_ONE
- **See** REGEX\_MULTIPPLICITY\_NEEDED
- **See** REGEX\_MULTIPPLICITY\_OPTIONNAL

*string* function `strhash($str, [$salt = true])` [line 1157]

### **Function Parameters:**

- *string* **\$str** the string to hash
- *mixed* **\$salt** If true, add the default salt : md5(\$str) if a string, use this string as the salt if false, do not use any salt

## Return a SHA256 hash of the \$str string [with a salt]

*string* function `strparse(&$content, [$forbidden_tags = array()], [$addslashes = true], $content)` [line 651]

### **Function Parameters:**

- *string* **\$content** Content to parse
- *string[]* **\$forbidden\_tags** List of the forbidden formatting tags
- *bool* **\$addslashes** if true, the parsed string will be escaped.
- **&\$content**

Parses a string with several default parameters. This methods exists to lighten the number of lines written.

- See [ContentParser](#)

*string* function `strprotect($var, [$html_protect = HTML_PROTECT], [$addslashes = ADDSLASHES_AUTO])` [*line 168*]

**Function Parameters:**

- *string* **\$var** Variable to protect.
- *bool* **\$html\_protect** HTML\_PROTECT if you don't accept the HTML code (it will be transformed by the corresponding HTML entities and won't be considerer by the web browsers). HTML\_UNPROTECT if you want to let them.
- *int* **\$addslashes** If you want to escape the quotes in the string, use ADDSLASHES\_FORCE, if you don't want, use the ADDSLASHES\_NONE constant. If you want to escape them only if they have not been escaped automatically by the magic quotes option, use the ADDSLASHES\_AUTO constant.

**Protects an input variable. Never trust user input!**

*string* function `strtotime($str, $date_format)` [*line 903*]

**Function Parameters:**

- *string* **\$str** Formatted date
- *string* **\$date\_format** Formatting pattern (DD for the day, MM for the month and YYYY for the year separated only by / characters).

**Converts a formatted date to the SQL date format.**

- **Deprecated**

*int* function strtotime(\$str, \$date\_format) [line 852]

**Function Parameters:**

- *string* **\$str** String to parse
- *string* **\$date\_format** Formatting pattern (d for day, m for month and y for year, for instance m/d/y)

### Parses a formatted date

- See [Date::Date\(\)](#)
- **Deprecated**

*string* function substr\_html(&\$str, \$start, [\$end = "], \$str) [line 273]

**Function Parameters:**

- *string* **\$str** The string you want to cut.
- *int* **\$start** If start is non-negative, the returned string will start at the start 'th position in string , counting from zero. For instance, in the string 'abcdef', the character at position 0 is 'a', the character at position 2 is 'c', and so forth. If start is negative, the returned string will start at the start 'th character from the end of string . If string is less than or equal to start characters long, FALSE will be returned.
- *int* **\$end** If length is given and is positive, the string returned will contain at most length characters beginning from start (depending on the length of string ).
- **&\$str**

**Cuts a string containing some HTML code which contains some HTML entities. The substr PHP function considers a HTML entity as several characters. This function allows you to consider them as only one character.**

TIMEZONE\_SITE = 1 [line 41]

TIMEZONE\_SYSTEM = 2 [line 42]

TIMEZONE\_USER = 3 [line 43]

*string* function to\_js\_string(\$string) [line 1277]

**Function Parameters:**

- *string* **\$string** A PHP string to convert to a JS one

*string* function `unparse(&$content, $content, $forbidden_tags, $addslashes)` [line 681]

**Function Parameters:**

- *string* **\$content** Content to unparse
- *string[]* **\$forbidden\_tags** List of the forbidden formatting tags
- *bool* **\$addslashes** if true, the unparsed string will be escaped.
- **&\$content**

**Unparses a string with several default parameters. This methods exists to lighten the number of lines written.**

- See [ContentUnparser](#)

*string* function `url($url, [$mod_rewrite = "], [$ampersand = '&'];)` [line 731]

**Function Parameters:**

- *string* **\$url** URL if the URL rewriting is disabled
- *string* **\$mod\_rewrite** URL if the URL rewriting is enabled
- *string* **\$ampersand** In a redirection you mustn't put the & HTML entity (&);. In this case set that parameter to &.

**Adds the session ID to an URL if the user doesn't accepts cookies. This functions allows you to generate an URL according to the site configuration concerning the URL rewriting.**

*string* function `url_encode_rewrite($string)` [line 766]

**Function Parameters:**

- *string* **\$string** String to encode.

**Prepares a string for it to be used in an URL (with only a-z, 0-9 and - characters).**

*string* function `wordwrap_html(&$str, $length, [$cut_char = '<br />'], [$cut = true], $str)` [line 257]

**Function Parameters:**

- *string* **\$str** The string to wrap.
- *int* **\$length** The number of characters you want in a line.
- *string* **\$cut\_char** The character to insert every \$length characters. The default value is ' ', the HTML carriage return tag.
- *bool* **\$cut** True if you accept that a word would be broken apart, false if you want to cut only on a blank character.
- **&\$str**

**Inserts a carriage return every \$length characters. It's equivalent to wordwrap PHP function but it can deal with the HTML entities. An entity is coded on several characters and the wordwrap function counts several characters for an entity whereas it represents only one character.**

# Appendices

# Appendix A - Class Trees

## Package members

### Authorizations

- [Authorizations](#)

### Group

- [Group](#)

### PrivateMsg

- [PrivateMsg](#)

### Session

- [Session](#)

### Uploads

- [Uploads](#)

### User

- [User](#)

## Package phpboost

## Package core

### Application

- [Application](#)

### BreadCrumb

- [BreadCrumb](#)

### Cache

- [Cache](#)

### Errors

- [Errors](#)

### MenuService

- [MenuService](#)

### Repository

- [Repository](#)

### StatsSaver

- [StatsSaver](#)

## Updates

- [Updates](#)

## Package menu

## Menu

- [Menu](#)
  - [ContentMenu](#)
  - [FeedMenu](#)
  - [LinksMenuElement](#)
    - [LinksMenu](#)
    - [LinksMenuLink](#)
  - [MiniMenu](#)
  - [ModuleMiniMenu](#)

## Package db

## Backup

- [Backup](#)

## Sql

- [Sql](#)

## Package builder

## FormBuilder

- [FormBuilder](#)

## FormField

- [FormField](#)
  - [FormCheckbox](#)
  - [FormCheckboxOption](#)
  - [FormFileUploader](#)
  - [FormHiddenField](#)
  - [FormRadioChoice](#)
  - [FormRadioChoiceOption](#)
  - [FormSelect](#)
  - [FormSelectOption](#)
  - [FormTextarea](#)
  - [FormTextEdit](#)

## FormFieldset

- [FormFieldset](#)

## Package modules

## ModuleInterface

- [ModuleInterface](#)

## ModulesDiscoveryService

- [ModulesDiscoveryService](#)

## PackagesManager

- [PackagesManager](#)

## Package content

### CategoriesManager

- [CategoriesManager](#)

### Comments

- [Comments](#)

### ContentEditor

- [ContentEditor](#)
  - [BBCodeEditor](#)
  - [TinyMCEEditor](#)

### ContentFormattingFactory

- [ContentFormattingFactory](#)

### Feed

- [Feed](#)
  - [ATOM](#)
  - [RSS](#)

## FeedData

- [FeedData](#)

## FeedItem

- [FeedItem](#)

## FeedsCat

- [FeedsCat](#)

## FeedsList

- [FeedsList](#)

## Note

- [Note](#)

## Parser

- [Parser](#)
  - [BBCodeHighlighter](#)
  - [ContentParser](#)
    - [BBCodeParser](#)
    - [TinyMCEParser](#)
  - [ContentSecondParser](#)
  - [ContentUnparser](#)
    - [BBCodeUnparser](#)
    - [TinyMCEUnparser](#)
  - [TemplateHighlighter](#)

## Search

- [Search](#)

## SiteMap

- [SiteMap](#)

## SiteMapElement

- [SiteMapElement](#)
  - [SiteMapLink](#)
  - [SiteMapSection](#)
    - [ModuleMap](#)

## SitemapExportConfig

- [SitemapExportConfig](#)

## Package events

## AdministratorAlertService

- [AdministratorAlertService](#)

## ContributionService

- [ContributionService](#)

## Event

- [Event](#)
  - [AdministratorAlert](#)
  - [Contribution](#)

## Package io

## FileSystemElement

- [FileSystemElement](#)
  - [File](#)
  - [Folder](#)

## Mail

- [Mail](#)

## Template

- [Template](#)

## Upload

- [Upload](#)

## Package util

### Bench

- [Bench](#)

### Captcha

- [Captcha](#)

### Date

- [Date](#)

### MiniCalendar

- [MiniCalendar](#)

### Pagination

- [Pagination](#)

### Stats

- [Stats](#)

### Url

- [Url](#)



# Index

## A

<a href="#">AdministratorAlert::build()</a>	135
<a href="#">AdministratorAlert::get_priority()</a>	135
<a href="#">AdministratorAlert::\$properties</a>	134
<a href="#">AdministratorAlert::\$priority</a>	134
<a href="#">AdministratorAlert</a>	134
<a href="#">AdministratorAlert::get_priority_name()</a>	135
<a href="#">AdministratorAlert::get_properties()</a>	135
<a href="#">AdministratorAlertService::compute_number_unread_alerts()</a>	136
<a href="#">AdministratorAlertService::delete_alert()</a>	136
<a href="#">AdministratorAlertService</a>	136
<a href="#">AdministratorAlert::set_properties()</a>	136
<a href="#">AdministratorAlert::set_priority()</a>	135
<a href="#">ADMINISTRATOR_ALERT_TYPE</a>	130
<a href="#">administrator_alert_service.class.php</a>	130
<a href="#">Application::load()</a>	101
<a href="#">administrator_alert.class.php</a>	129
<a href="#">Application::get_warning_level()</a>	101
<a href="#">Application::get_warning()</a>	101
<a href="#">Application::get_version()</a>	100
<a href="#">ADMIN_ALERT_HIGH_PRIORITY</a>	129
<a href="#">ADMIN_ALERT_LOW_PRIORITY</a>	129
<a href="#">ADMIN_ALERT_VERY_HIGH_PRIORITY</a>	129
<a href="#">ADMIN_ALERT_VERY_LOW_PRIORITY</a>	129
<a href="#">ADMIN_ALERT_STATUS_UNREAD</a>	129
<a href="#">ADMIN_ALERT_STATUS_PROCESSED</a>	129
<a href="#">ADMIN_ALERT_MEDIUM_PRIORITY</a>	129
<a href="#">AdministratorAlertService::find_by_criteria()</a>	137
<a href="#">AdministratorAlertService::find_by_id()</a>	137
<a href="#">Authorizations::capture_and_shift_bit_auth()</a>	188
<a href="#">Authorizations::check_auth()</a>	188
<a href="#">Authorizations::build_auth_array_from_form()</a>	188
<a href="#">Authorizations::auth_array_simple()</a>	187
<a href="#">Authorizations</a>	187
<a href="#">Authorizations::generate_select()</a>	189
<a href="#">Authorizations::merge_auth()</a>	189
<a href="#">ADDSLASHES_NONE</a>	264
<a href="#">array_combine()</a>	264
<a href="#">ADDSLASHES_FORCE</a>	264
<a href="#">ADDSLASHES_AUTO</a>	264
<a href="#">ACCES_DENIED</a>	228
<a href="#">ADMIN_NO_CHECK</a>	185
<a href="#">AUTOCONNECT</a>	184
<a href="#">AdministratorAlertService::get_number_unread_alerts()</a>	138
<a href="#">AdministratorAlertService::save_alert()</a>	139

<a href="#">AdministratorAlertService::get number alerts()</a>	138
<a href="#">AdministratorAlertService::get all alerts()</a>	138
<a href="#">AdministratorAlertService::find by identifier()</a>	137
<a href="#">AUTO LOAD FREQUENT VARS</a>	156
<a href="#">ADD</a>	171
<a href="#">ADMIN_NOAUTH_DEFAULT</a>	182
<a href="#">ALREADY HASHED</a>	184
<a href="#">AUTH_PARENT_PRIORITY</a>	181
<a href="#">AUTH_CHILD_PRIORITY</a>	181
<a href="#">authorizations.class.php</a>	181
<a href="#">Application::get update url()</a>	100
<a href="#">Application::get type()</a>	100
<a href="#">Application::\$download_url</a>	97
<a href="#">Application::\$id</a>	97
<a href="#">Application::\$description</a>	97
<a href="#">Application::\$compatibility_min</a>	97
<a href="#">Application::\$compatibility_max</a>	97
<a href="#">Application::\$improvements</a>	98
<a href="#">Application::\$language</a>	98
<a href="#">Application::\$priority</a>	98
<a href="#">Application::\$pubdate</a>	98
<a href="#">Application::\$new_features</a>	98
<a href="#">Application::\$name</a>	98
<a href="#">Application::\$localized language</a>	98
<a href="#">Application::\$bug_corrections</a>	97
<a href="#">Application::\$authors</a>	97
<a href="#">ATOM</a>	76
<a href="#">ATOM::load file()</a>	76
<a href="#">atom.class.php</a>	73
<a href="#">ADD SLASHES</a>	44
<a href="#">AJAX_MODE</a>	20
<a href="#">application.class.php</a>	91
<a href="#">APPLICATION_TYPE_KERNEL</a>	91
<a href="#">ARCHIVE ERROR</a>	93
<a href="#">Application</a>	97
<a href="#">ARCHIVE ALL ERRORS</a>	93
<a href="#">APPLICATION_TYPE_TEMPLATE</a>	91
<a href="#">APPLICATION_TYPE_MODULE</a>	91
<a href="#">Application::\$repository</a>	98
<a href="#">Application::\$security_improvements</a>	98
<a href="#">Application::get language()</a>	100
<a href="#">Application::get localized language()</a>	100
<a href="#">Application::get improvements()</a>	99
<a href="#">Application::get identifier()</a>	99
<a href="#">Application::get id()</a>	99
<a href="#">Application::get name()</a>	100
<a href="#">Application::get new_features()</a>	100
<a href="#">Application::get security_improvements()</a>	100
<a href="#">Application::get security_update()</a>	100
<a href="#">Application::get repository()</a>	100
<a href="#">Application::get pubdate()</a>	100
<a href="#">Application::get priority()</a>	100
<a href="#">Application::get download_url()</a>	99

<a href="#">Application::get_description()</a>	99
<a href="#">Application::\$version</a>	98
<a href="#">Application::\$warning</a>	98
<a href="#">Application::\$update_url</a>	98
<a href="#">Application::\$type</a>	98
<a href="#">Application::\$security_update</a>	98
<a href="#">Application::\$warning_level</a>	98
<a href="#">Application::check_compatibility()</a>	99
<a href="#">Application::get_compatibility_min()</a>	99
<a href="#">Application::get_compatibility_max()</a>	99
<a href="#">Application::get_bug_corrections()</a>	99
<a href="#">Application::get_authors()</a>	99
<a href="#">ADD THIS CATEGORY IN LIST</a>	20

## B

<a href="#">Backup::get_tables_properties_list()</a>	118
<a href="#">Backup::get_tables_number()</a>	118
<a href="#">Backup::list_db_tables()</a>	118
<a href="#">BLOCK POSITION ALL</a>	204
<a href="#">BLOCK POSITION BOTTOM CENTRAL</a>	204
<a href="#">Backup::get_tables_list()</a>	118
<a href="#">Backup::get_script()</a>	118
<a href="#">Backup::extract_table_structure()</a>	117
<a href="#">Backup::export_file()</a>	117
<a href="#">Backup::generate_create_table_query()</a>	117
<a href="#">Backup::generate_drop_table_query()</a>	117
<a href="#">Backup::generate_insert_values_query()</a>	118
<a href="#">BLOCK POSITION FOOTER</a>	204
<a href="#">BLOCK POSITION HEADER</a>	204
<a href="#">Bench::\$start</a>	244
<a href="#">Bench::\$duration</a>	244
<a href="#">Bench::get_microtime()</a>	245
<a href="#">Bench::stop()</a>	245
<a href="#">Bench::to_string()</a>	245
<a href="#">Bench</a>	244
<a href="#">BLOCK POSITION TOP FOOTER</a>	204
<a href="#">BLOCK POSITION NOT_ENABLED</a>	204
<a href="#">BLOCK POSITION LEFT</a>	204
<a href="#">BLOCK POSITION RIGHT</a>	204
<a href="#">BLOCK POSITION SUB HEADER</a>	204
<a href="#">BLOCK POSITION TOP CENTRAL</a>	204
<a href="#">Backup::concatenate_to_query()</a>	117
<a href="#">Backup::\$tables</a>	116
<a href="#">BBCODE LANGUAGE</a>	43
<a href="#">BBCODE TAG COLOR</a>	42
<a href="#">BBCodeHighlighter</a>	45
<a href="#">BBCodeHighlighter::parse()</a>	45
<a href="#">BBCodeParser</a>	46
<a href="#">BBCODE PARAM NAME COLOR</a>	42
<a href="#">BBCODE PARAM COLOR</a>	42
<a href="#">bbcode_highlighter.class.php</a>	42

<a href="#">BBCodeEditor::display()</a>	39
<a href="#">BBCODE HIGHLIGHTER BLOCK CODE</a>	42
<a href="#">BBCODE HIGHLIGHTER INLINE CODE</a>	42
<a href="#">BBCODE LIST ITEM COLOR</a>	42
<a href="#">BBCodeParser::clear_html_br()</a>	46
<a href="#">BBCodeParser::parse()</a>	46
<a href="#">BreadCrumb::remove_last()</a>	102
<a href="#">BreadCrumb::display()</a>	102
<a href="#">BreadCrumb::reverse()</a>	102
<a href="#">Backup</a>	116
<a href="#">Backup::\$backup_script</a>	116
<a href="#">BreadCrumb::clean()</a>	102
<a href="#">BreadCrumb::add()</a>	102
<a href="#">BBCodeUnparser</a>	46
<a href="#">BBCodeUnparser::parse()</a>	47
<a href="#">BreadCrumb</a>	101
<a href="#">BreadCrumb::\$array_links</a>	101
<a href="#">BBCodeEditor</a>	39

## C

<a href="#">Contribution::get_status_name()</a>	143
<a href="#">Contribution::get_poster_login()</a>	142
<a href="#">Contribution::get_poster_id()</a>	142
<a href="#">Contribution::set_auth()</a>	143
<a href="#">Contribution::set_description()</a>	143
<a href="#">Contribution::set_fixing_date()</a>	143
<a href="#">Contribution::set_fixer_id()</a>	143
<a href="#">Contribution::get_module_name()</a>	142
<a href="#">Contribution::get_module()</a>	142
<a href="#">Contribution::get_auth()</a>	142
<a href="#">Contribution::build()</a>	141
<a href="#">Contribution::get_description()</a>	142
<a href="#">Contribution::get_fixer_id()</a>	142
<a href="#">Contribution::get_fixing_date()</a>	142
<a href="#">Contribution::get_fixer_login()</a>	142
<a href="#">Contribution::set_module()</a>	143
<a href="#">Contribution::set_poster_id()</a>	144
<a href="#">constructor Event::Event()</a>	149
<a href="#">ContributionService::save_contribution()</a>	147
<a href="#">CRLF</a>	155
<a href="#">CHECK_EXIST</a>	157
<a href="#">constructor Template::Template()</a>	165
<a href="#">constructor Mail::Mail()</a>	160
<a href="#">ContributionService::get_all_contributions()</a>	146
<a href="#">ContributionService::generate_cache()</a>	146
<a href="#">ContributionService</a>	144
<a href="#">Contribution::set_status()</a>	144
<a href="#">ContributionService::compute_number_contrib_for_each_profile()</a>	145
<a href="#">ContributionService::delete_contribution()</a>	145
<a href="#">ContributionService::find_by_id()</a>	146
<a href="#">ContributionService::find_by_criteria()</a>	145

<a href="#">constructor Contribution::Contribution()</a>	141
<a href="#">Contribution::\$poster_login</a>	141
<a href="#">Cache::write()</a>	104
<a href="#">Cache::load()</a>	104
<a href="#">constructor Errors::Errors()</a>	105
<a href="#">constructor Repository::Repository()</a>	112
<a href="#">CONNECTED TO DATABASE</a>	115
<a href="#">constructor Updates::Updates()</a>	113
<a href="#">Cache::generate_module_file()</a>	104
<a href="#">Cache::generate_file()</a>	103
<a href="#">Cache::\$files</a>	103
<a href="#">Cache</a>	102
<a href="#">constructor Cache::Cache()</a>	103
<a href="#">Cache::delete_file()</a>	103
<a href="#">Cache::generate_all_modules()</a>	103
<a href="#">Cache::generate_all_files()</a>	103
<a href="#">CONNECTION FAILED</a>	115
<a href="#">constructor Backup::Backup()</a>	116
<a href="#">Contribution::\$fixer_id</a>	140
<a href="#">Contribution::\$description</a>	140
<a href="#">Contribution::\$fixer_login</a>	140
<a href="#">Contribution::\$fixing_date</a>	140
<a href="#">Contribution::\$poster_id</a>	141
<a href="#">Contribution::\$module</a>	141
<a href="#">Contribution::\$auth</a>	140
<a href="#">Contribution</a>	139
<a href="#">contribution.class.php</a>	131
<a href="#">constructor Sql::Sql()</a>	120
<a href="#">CONTRIBUTION_AUTH_BIT</a>	131
<a href="#">contribution_service.class.php</a>	132
<a href="#">constructor AdministratorAlert::AdministratorAlert()</a>	135
<a href="#">CONTRIBUTION_TYPE</a>	132
<a href="#">constructor Upload::Upload()</a>	169
<a href="#">CLOSEFILE</a>	171
<a href="#">Captcha::\$width</a>	246
<a href="#">Captcha::\$instance</a>	246
<a href="#">Captcha::\$height</a>	246
<a href="#">constructor Captcha::Captcha()</a>	246
<a href="#">Captcha::display()</a>	246
<a href="#">Captcha::is_available()</a>	246
<a href="#">Captcha::display_form()</a>	246
<a href="#">Captcha::\$gd_loaded</a>	246
<a href="#">Captcha::\$font</a>	246
<a href="#">CAPTCHA VERY HARD</a>	238
<a href="#">CAPTCHA VERY EASY</a>	238
<a href="#">constructor Bench::Bench()</a>	245
<a href="#">Captcha</a>	245
<a href="#">Captcha::\$difficulty</a>	246
<a href="#">Captcha::\$code</a>	245
<a href="#">Captcha::is_valid()</a>	246
<a href="#">Captcha::js_require()</a>	246
<a href="#">constructor Stats::Stats()</a>	256
<a href="#">constructor Pagination::Pagination()</a>	254

<a href="#">constructor Url::Url()</a>	257
<a href="#">check_mail()</a>	264
<a href="#">CLASS_IMPORT</a>	265
<a href="#">check_nbr_links()</a>	264
<a href="#">constructor MiniCalendar::MiniCalendar()</a>	252
<a href="#">constructor Date::Date()</a>	248
<a href="#">Captcha::set_font()</a>	247
<a href="#">Captcha::set_difficulty()</a>	247
<a href="#">Captcha::set_height()</a>	247
<a href="#">Captcha::set_instance()</a>	247
<a href="#">Captcha::update_instance()</a>	248
<a href="#">Captcha::set_width()</a>	247
<a href="#">CAPTCHA_NORMAL</a>	238
<a href="#">CAPTCHA_HARD</a>	238
<a href="#">ContentMenu</a>	210
<a href="#">CONTENT_MENU_CLASS</a>	210
<a href="#">ContentMenu::\$content</a>	210
<a href="#">ContentMenu::\$display_title</a>	210
<a href="#">ContentMenu::cache_export()</a>	211
<a href="#">constructor ContentMenu::ContentMenu()</a>	210
<a href="#">content_menu.class.php</a>	210
<a href="#">constructor Menu::Menu()</a>	206
<a href="#">constructor FileSystemElement::FileSystemElement()</a>	176
<a href="#">constructor File::File()</a>	173
<a href="#">constructor Folder::Folder()</a>	178
<a href="#">CHECK_PM_BOX</a>	183
<a href="#">constructor User::User()</a>	199
<a href="#">constructor Group::Group()</a>	190
<a href="#">ContentMenu::display()</a>	211
<a href="#">ContentMenu::get_content()</a>	211
<a href="#">CONFIG_CONFLICT</a>	229
<a href="#">constructor ModuleMiniMenu::ModuleMiniMenu()</a>	226
<a href="#">constructor ModuleInterface::ModuleInterface()</a>	232
<a href="#">constructor ModulesDiscoveryService::ModulesDiscoveryService()</a>	234
<a href="#">CAPTCHA_EASY</a>	238
<a href="#">captcha.class.php</a>	238
<a href="#">constructor MiniMenu::MiniMenu()</a>	225
<a href="#">constructor LinksMenuLink::LinksMenuLink()</a>	224
<a href="#">ContentMenu::set_content()</a>	211
<a href="#">ContentMenu::get_display_title()</a>	211
<a href="#">ContentMenu::set_display_title()</a>	211
<a href="#">constructor FeedMenu::FeedMenu()</a>	213
<a href="#">constructor LinksMenuElement::LinksMenuElement()</a>	221
<a href="#">constructor LinksMenu::LinksMenu()</a>	218
<a href="#">constructor BreadCrumb::BreadCrumb()</a>	102
<a href="#">constructor Application::Application()</a>	99
<a href="#">Comments::\$idprov</a>	31
<a href="#">Comments::\$idcom</a>	31
<a href="#">Comments</a>	31
<a href="#">Comments::\$is_kernel_script</a>	31
<a href="#">Comments::\$lock_com</a>	31
<a href="#">Comments::\$nbr_com</a>	31
<a href="#">Comments::\$module_folder</a>	31

<a href="#">CategoriesManager::set_display_config()</a>	30
<a href="#">CategoriesManager::move_into_another()</a>	30
<a href="#">CategoriesManager::check_error()</a>	29
<a href="#">CategoriesManager::check_display_config()</a>	29
<a href="#">CategoriesManager::compute_heritated_auth()</a>	29
<a href="#">CategoriesManager::delete()</a>	29
<a href="#">CategoriesManager::move()</a>	30
<a href="#">CategoriesManager::get_feeds_list()</a>	30
<a href="#">Comments::\$path</a>	31
<a href="#">Comments::\$script</a>	31
<a href="#">Comments::is_loaded()</a>	33
<a href="#">Comments::get_attribute()</a>	33
<a href="#">Comments::lock()</a>	33
<a href="#">Comments::set_arg()</a>	34
<a href="#">constructor Note::Note()</a>	35
<a href="#">Comments::update()</a>	34
<a href="#">Comments::display()</a>	33
<a href="#">Comments::delete_all()</a>	33
<a href="#">Comments::\$vars</a>	32
<a href="#">Comments::\$sql_table</a>	32
<a href="#">constructor Comments::Comments()</a>	32
<a href="#">Comments::add()</a>	32
<a href="#">Comments::del()</a>	33
<a href="#">Comments::com_display_link()</a>	32
<a href="#">CategoriesManager::change_visibility()</a>	28
<a href="#">CategoriesManager::build_select_form()</a>	28
<a href="#">constructor FormSelectOption::FormSelectOption()</a>	16
<a href="#">constructor FormSelect::FormSelect()</a>	15
<a href="#">constructor FormTextarea::FormTextarea()</a>	17
<a href="#">constructor FormTextEdit::FormTextEdit()</a>	18
<a href="#">CATEGORY_DOES_NOT_EXIST</a>	20
<a href="#">categories_manager.class.php</a>	20
<a href="#">constructor FormRadioChoiceOption::FormRadioChoiceOption()</a>	15
<a href="#">constructor FormRadioChoice::FormRadioChoice()</a>	14
<a href="#">constructor FormCheckboxOption::FormCheckboxOption()</a>	8
<a href="#">constructor FormCheckbox::FormCheckbox()</a>	7
<a href="#">constructor FormField::FormField()</a>	9
<a href="#">constructor FormFieldset::FormFieldset()</a>	10
<a href="#">constructor FormHiddenField::FormHiddenField()</a>	13
<a href="#">constructor FormFileUploader::FormFileUploader()</a>	12
<a href="#">CAT_UNVISIBLE</a>	20
<a href="#">CAT_VISIBLE</a>	20
<a href="#">constructor CategoriesManager::CategoriesManager()</a>	26
<a href="#">CategoriesManager::\$stable</a>	26
<a href="#">CategoriesManager::add()</a>	26
<a href="#">CategoriesManager::build_administration_interface()</a>	27
<a href="#">CategoriesManager::build_parents_id_list()</a>	28
<a href="#">CategoriesManager::build_children_id_list()</a>	27
<a href="#">CategoriesManager::\$errors</a>	26
<a href="#">CategoriesManager::\$display_config</a>	26
<a href="#">CACHE_TIME</a>	23
<a href="#">comments.class.php</a>	21
<a href="#">CACHE_TIMES_USED</a>	23

<a href="#">CategoriesManager</a>	24
<a href="#">CategoriesManager::\$cache_var</a>	25
<a href="#">CategoriesManager::\$cache_file_name</a>	25
<a href="#">constructor Search::Search()</a>	37
<a href="#">constructor BBCodeEditor::BBCodeEditor()</a>	39
<a href="#">constructor Parser::Parser()</a>	53
<a href="#">constructor ContentUnparser::ContentUnparser()</a>	52
<a href="#">ContentUnparser</a>	52
<a href="#">constructor TemplateHighlighter::TemplateHighlighter()</a>	55
<a href="#">constructor TinyMCEParser::TinyMCEParser()</a>	56
<a href="#">constructor ModuleMap::ModuleMap()</a>	58
<a href="#">constructor TinyMCEUnparser::TinyMCEUnparser()</a>	56
<a href="#">ContentSecondParser::parse()</a>	51
<a href="#">ContentSecondParser::export_html_text()</a>	51
<a href="#">ContentParser::parse()</a>	50
<a href="#">ContentParser::get_forbidden_tags()</a>	50
<a href="#">ContentParser::set_forbidden_tags()</a>	50
<a href="#">ContentParser::set_html_auth()</a>	50
<a href="#">constructor ContentSecondParser::ContentSecondParser()</a>	51
<a href="#">ContentSecondParser</a>	51
<a href="#">constructor SiteMap::SiteMap()</a>	60
<a href="#">constructor SiteMapElement::SiteMapElement()</a>	62
<a href="#">cache.class.php</a>	92
<a href="#">constructor RSS::RSS()</a>	89
<a href="#">CHECK ALL UPDATES</a>	96
<a href="#">CHECK KERNEL</a>	96
<a href="#">CHECK THEMES</a>	96
<a href="#">CHECK MODULES</a>	96
<a href="#">constructor FeedsList::FeedsList()</a>	88
<a href="#">constructor FeedsCat::FeedsCat()</a>	87
<a href="#">constructor SiteMapSection::SiteMapSection()</a>	70
<a href="#">constructor SitemapExportConfig::SitemapExportConfig()</a>	64
<a href="#">constructor ATOM::ATOM()</a>	76
<a href="#">constructor Feed::Feed()</a>	77
<a href="#">constructor FeedItem::FeedItem()</a>	84
<a href="#">constructor FeedData::FeedData()</a>	81
<a href="#">constructor ContentParser::ContentParser()</a>	50
<a href="#">ContentParser::\$tag</a>	50
<a href="#">ContentEditor::set_identifier()</a>	40
<a href="#">ContentEditor::set_forbidden_tags()</a>	40
<a href="#">ContentEditor::set_template()</a>	40
<a href="#">constructor TinyMCEEditor::TinyMCEEditor()</a>	41
<a href="#">constructor BBCodeHighlighter::BBCodeHighlighter()</a>	45
<a href="#">content formatting factory.class.php</a>	43
<a href="#">ContentEditor::get_template()</a>	40
<a href="#">ContentEditor::get_forbidden_tags()</a>	40
<a href="#">ContentEditor::\$forbidden_tags</a>	40
<a href="#">ContentEditor</a>	39
<a href="#">ContentEditor::\$identifier</a>	40
<a href="#">ContentEditor::\$language_type</a>	40
<a href="#">constructor ContentEditor::ContentEditor()</a>	40
<a href="#">ContentEditor::\$template</a>	40
<a href="#">constructor BBCodeParser::BBCodeParser()</a>	46

<a href="#">constructor BBCodeUnparser::BBCodeUnparser()</a>	47
<a href="#">ContentFormattingFactory::get_user_editor()</a>	48
<a href="#">ContentFormattingFactory::get_unparser()</a>	48
<a href="#">ContentFormattingFactory::set_language()</a>	49
<a href="#">ContentParser</a>	49
<a href="#">ContentParser::\$html_auth</a>	50
<a href="#">ContentParser::\$forbidden_tags</a>	49
<a href="#">ContentFormattingFactory::get_second_parser()</a>	48
<a href="#">ContentFormattingFactory::get_parser()</a>	48
<a href="#">ContentFormattingFactory::\$language_type</a>	47
<a href="#">ContentFormattingFactory</a>	47
<a href="#">constructor ContentFormattingFactory::ContentFormattingFactory()</a>	47
<a href="#">ContentFormattingFactory::get_available_tags()</a>	48
<a href="#">ContentFormattingFactory::get_language()</a>	48
<a href="#">ContentFormattingFactory::get_editor()</a>	48
<a href="#">constructor FormBuilder::FormBuilder()</a>	5

## D

<a href="#">DRAW VALUES</a>	240
<a href="#">Date</a>	248
<a href="#">Date::\$timestamp</a>	248
<a href="#">Date::check_date()</a>	249
<a href="#">DRAW LEGEND</a>	240
<a href="#">DATE YEAR MONTH DAY HOUR MINUTE SECOND</a>	239
<a href="#">DATE RFC3339 F</a>	239
<a href="#">DATE RFC3339 FORMAT</a>	239
<a href="#">DATE TIMESTAMP</a>	239
<a href="#">DATE YEAR MONTH DAY</a>	239
<a href="#">Date::format()</a>	250
<a href="#">Date::get_day()</a>	250
<a href="#">Date::to_date()</a>	251
<a href="#">delete file()</a>	265
<a href="#">display comments()</a>	265
<a href="#">display editor()</a>	265
<a href="#">Date::get_year()</a>	251
<a href="#">Date::get_timestamp()</a>	251
<a href="#">Date::get_hours()</a>	250
<a href="#">Date::get_minutes()</a>	250
<a href="#">Date::get_month()</a>	250
<a href="#">Date::get_seconds()</a>	251
<a href="#">DATE RFC822 FORMAT</a>	239
<a href="#">DATE RFC822 F</a>	239
<a href="#">DEFAULT FEED NAME</a>	74
<a href="#">DEFAULT RSS TEMPLATE</a>	76
<a href="#">DISPLAY ALL ERROR</a>	93
<a href="#">DBTYPE</a>	115
<a href="#">DEFAULT ATOM TEMPLATE</a>	73
<a href="#">DO NOT ADD SLASHES</a>	44
<a href="#">DISPLAYING CONFIGURATION NOT SET</a>	20
<a href="#">DO NOT ADD THIS CATEGORY IN LIST</a>	20
<a href="#">DO NOT LOAD CACHE</a>	20

<a href="#">DEFAULT LANGUAGE</a>	43
<a href="#">DB_NO_CONNECT</a>	115
<a href="#">DO NOT AUTO LOAD FREQUENT VARS</a>	156
<a href="#">DATE_FORMAT_SHORT</a>	239
<a href="#">DATE_FORMAT_TINY</a>	239
<a href="#">DATE_FROM_STRING</a>	239
<a href="#">DATE_NOW</a>	239
<a href="#">DATE_FORMAT_LONG</a>	239
<a href="#">DATE_FORMAT</a>	239
<a href="#">DELETE_ON_ERROR</a>	157
<a href="#">DEL_PM_CONVERS</a>	183
<a href="#">DO NOT GENERATE CACHE AFTER THE OPERATION</a>	229
<a href="#">date.class.php</a>	239
<a href="#">DEBUG_MODE</a>	20

## E

<a href="#">Event::get_fixing_url()</a>	150
<a href="#">Event::get_entitled()</a>	150
<a href="#">Event::get_id()</a>	150
<a href="#">Event::get_identifier()</a>	150
<a href="#">Event::get_must_regenerate_cache()</a>	150
<a href="#">Event::get_id_in_module()</a>	150
<a href="#">Event::get_creation_date()</a>	150
<a href="#">Event::build()</a>	149
<a href="#">Event::\$identifier</a>	149
<a href="#">Event::\$id</a>	148
<a href="#">Event::\$id_in_module</a>	149
<a href="#">Event::\$must_regenerate_cache</a>	149
<a href="#">Event::\$type</a>	149
<a href="#">Event::get_status()</a>	150
<a href="#">Event::get_status_name()</a>	151
<a href="#">Event::set_status()</a>	152
<a href="#">Event::set_must_regenerate_cache()</a>	152
<a href="#">Event::set_type()</a>	153
<a href="#">ERASE</a>	171
<a href="#">EMPTY_FOLDER</a>	185
<a href="#">Event::set_id_in_module()</a>	152
<a href="#">Event::set_identifier()</a>	152
<a href="#">Event::set_creation_date()</a>	151
<a href="#">Event::get_type()</a>	151
<a href="#">Event::set_entitled()</a>	151
<a href="#">Event::set_fixing_url()</a>	151
<a href="#">Event::set_id()</a>	151
<a href="#">Event::\$fixing_url</a>	148
<a href="#">Event::\$entitled</a>	148
<a href="#">Errors::\$redirect</a>	105
<a href="#">Errors::\$personal_tpl</a>	105
<a href="#">Errors::\$template</a>	105
<a href="#">Errors::display()</a>	105
<a href="#">Errors::get_errno_class()</a>	105
<a href="#">Errors::\$archive_all</a>	105

<a href="#">Errors</a>	104
<a href="#">ERROR_UNKNOWN_MOTION</a>	20
<a href="#">ERROR_CAT_IS_AT_TOP</a>	20
<a href="#">ERROR_GETTING_CACHE</a>	74
<a href="#">errors.class.php</a>	93
<a href="#">E_STRICT</a>	93
<a href="#">Errors::get_last_error_log()</a>	106
<a href="#">Errors::handler()</a>	106
<a href="#">EVENT_STATUS_UNREAD</a>	133
<a href="#">EVENT_STATUS_PROCESSED</a>	133
<a href="#">Event</a>	147
<a href="#">Event::\$creation_date</a>	147
<a href="#">Event::\$current_status</a>	148
<a href="#">EVENT_STATUS_BEING_PROCESSED</a>	133
<a href="#">event.class.php</a>	133
<a href="#">Errors::set_default_template()</a>	106
<a href="#">Errors::handler_php()</a>	106
<a href="#">Errors::set_template()</a>	106
<a href="#">ERRORS_MANAGEMENT_BY_RETURN</a>	115
<a href="#">EXPLICIT_ERRORS_MANAGEMENT</a>	115
<a href="#">ERROR_CAT_IS_AT_BOTTOM</a>	20

## F

<a href="#">FeedItem::set_auth()</a>	84
<a href="#">FeedItem::get_title()</a>	84
<a href="#">FeedItem::set_date()</a>	85
<a href="#">FeedItem::set_desc()</a>	85
<a href="#">FeedItem::set_image_url()</a>	85
<a href="#">FeedItem::set_guid()</a>	85
<a href="#">FeedItem::get_link()</a>	84
<a href="#">FeedItem::get_image_url()</a>	84
<a href="#">FeedItem::get_date()</a>	84
<a href="#">FeedItem::get_auth()</a>	84
<a href="#">FeedItem::get_date_rfc822()</a>	84
<a href="#">FeedItem::get_date_rfc3339()</a>	84
<a href="#">FeedItem::get_guid()</a>	84
<a href="#">FeedItem::get_desc()</a>	84
<a href="#">FeedItem::set_link()</a>	85
<a href="#">FeedItem::set_title()</a>	86
<a href="#">FeedsCat::get_module_id()</a>	87
<a href="#">FeedsCat::get_children()</a>	87
<a href="#">FeedsCat::get_url()</a>	87
<a href="#">FeedsList</a>	88
<a href="#">FeedsList::add_feed()</a>	88
<a href="#">FeedsList::\$list</a>	88
<a href="#">FeedsCat::get_category_name()</a>	87
<a href="#">FeedsCat::get_category_id()</a>	87
<a href="#">FeedsCat::\$cat_name</a>	86
<a href="#">FeedsCat</a>	86
<a href="#">FeedsCat::\$children</a>	86
<a href="#">FeedsCat::\$id</a>	86

<a href="#">FeedsCat::add_child()</a>	87
<a href="#">FeedsCat::\$module_id</a>	87
<a href="#">FeedItem::\$title</a>	84
<a href="#">FeedItem::\$link</a>	84
<a href="#">FeedData::get_host()</a>	82
<a href="#">FeedData::get_desc()</a>	82
<a href="#">FeedData::get_items()</a>	82
<a href="#">FeedData::get_lang()</a>	82
<a href="#">FeedData::get_title()</a>	82
<a href="#">FeedData::get_link()</a>	82
<a href="#">FeedData::get_date_rfc3339()</a>	82
<a href="#">FeedData::get_date_rfc822()</a>	81
<a href="#">FeedData::\$lang</a>	81
<a href="#">FeedData::\$items</a>	81
<a href="#">FeedData::\$link</a>	81
<a href="#">FeedData::\$title</a>	81
<a href="#">FeedData::get_date()</a>	81
<a href="#">FeedData::add_item()</a>	81
<a href="#">FeedData::serialize()</a>	82
<a href="#">FeedData::set_auth_bit()</a>	82
<a href="#">FeedItem::\$auth</a>	84
<a href="#">FeedItem</a>	83
<a href="#">FeedItem::\$date</a>	84
<a href="#">FeedItem::\$desc</a>	84
<a href="#">FeedItem::\$image_url</a>	84
<a href="#">FeedItem::\$guid</a>	84
<a href="#">FeedData::subitems()</a>	83
<a href="#">FeedData::set_title()</a>	83
<a href="#">FeedData::set_desc()</a>	82
<a href="#">FeedData::set_date()</a>	82
<a href="#">FeedData::set_host()</a>	82
<a href="#">FeedData::set_lang()</a>	83
<a href="#">FeedData::set_link()</a>	83
<a href="#">FeedsList::get_feeds_list()</a>	88
<a href="#">file.class.php</a>	171
<a href="#">FEED MENU CLASS</a>	212
<a href="#">feed_menu.class.php</a>	212
<a href="#">FeedMenu</a>	212
<a href="#">FeedMenu::\$begin_at</a>	212
<a href="#">FeedMenu::\$module_id</a>	212
<a href="#">FeedMenu::\$category</a>	212
<a href="#">Folder::open()</a>	179
<a href="#">Folder::get_folders()</a>	179
<a href="#">Folder::\$folders</a>	178
<a href="#">Folder::\$files</a>	178
<a href="#">Folder::delete()</a>	178
<a href="#">Folder::get_all_content()</a>	179
<a href="#">Folder::get_first_folder()</a>	179
<a href="#">Folder::get_files()</a>	179
<a href="#">FeedMenu::\$name</a>	212
<a href="#">FeedMenu::\$number</a>	212
<a href="#">FUNCTIONNALITY NOT IMPLEMENTED</a>	228
<a href="#">FeedMenu::set_name()</a>	214

<a href="#">FRANKLINBC TTF</a>	240
<a href="#">functions.inc.php</a>	264
<a href="#">find_require_dir()</a>	266
<a href="#">file_get_contents_emulate()</a>	266
<a href="#">FeedMenu::set_module_id()</a>	214
<a href="#">FeedMenu::set_cat()</a>	213
<a href="#">FeedMenu::cache_export()</a>	213
<a href="#">FeedMenu::\$url</a>	212
<a href="#">FeedMenu::display()</a>	213
<a href="#">FeedMenu::get_module_id()</a>	213
<a href="#">FeedMenu::get_url()</a>	213
<a href="#">FeedMenu::get_template()</a>	213
<a href="#">Folder</a>	178
<a href="#">FileSystemElement::write()</a>	177
<a href="#">File::flush()</a>	174
<a href="#">File::finclude()</a>	173
<a href="#">File::frequire()</a>	174
<a href="#">File::get_contents()</a>	174
<a href="#">File::get_last_modification_date()</a>	174
<a href="#">File::get_last_access_date()</a>	174
<a href="#">File::delete()</a>	173
<a href="#">File::close()</a>	173
<a href="#">File</a>	172
<a href="#">file_system_element.class.php</a>	172
<a href="#">File::\$contents</a>	172
<a href="#">File::\$fd</a>	172
<a href="#">File::\$mode</a>	173
<a href="#">File::\$lines</a>	173
<a href="#">File::get_lines()</a>	174
<a href="#">File::is_open()</a>	175
<a href="#">FileSystemElement::delete()</a>	177
<a href="#">FileSystemElement::change_chmod()</a>	176
<a href="#">FileSystemElement::exists()</a>	177
<a href="#">FileSystemElement::get()</a>	177
<a href="#">FileSystemElement::open()</a>	177
<a href="#">FileSystemElement::get_name()</a>	177
<a href="#">FileSystemElement::\$path</a>	176
<a href="#">FileSystemElement::\$is_open</a>	176
<a href="#">File::open()</a>	175
<a href="#">File::lock()</a>	175
<a href="#">File::unlock()</a>	175
<a href="#">File::write()</a>	175
<a href="#">FileSystemElement</a>	175
<a href="#">FeedData::\$host</a>	81
<a href="#">FeedData::\$desc</a>	81
<a href="#">FormField::\$field_name</a>	9
<a href="#">FormField::\$field_id</a>	8
<a href="#">FormField::\$field_on_blur</a>	9
<a href="#">FormField::\$field_required</a>	9
<a href="#">FormField::\$field_sub_title</a>	9
<a href="#">FormField::\$field_required_alert</a>	9
<a href="#">FormField::\$field_errors</a>	8
<a href="#">FormField::\$field_css_class</a>	8

<a href="#">FormCheckboxOption</a>	7
<a href="#">FormCheckbox::display()</a>	7
<a href="#">FormCheckboxOption::\$option_checked</a>	7
<a href="#">FormCheckboxOption::\$option_title</a>	8
<a href="#">FormField</a>	8
<a href="#">FormCheckboxOption::display()</a>	8
<a href="#">FormField::\$field_title</a>	9
<a href="#">FormField::\$field_value</a>	9
<a href="#">FormFieldset::\$fieldset_title</a>	10
<a href="#">FormFieldset::\$fieldset_fields</a>	10
<a href="#">FormFieldset::add_field()</a>	11
<a href="#">FormFieldset::display()</a>	11
<a href="#">FormFieldset::get_display_required()</a>	11
<a href="#">FormFieldset::display_preview_button()</a>	11
<a href="#">FormFieldset::\$fieldset_errors</a>	10
<a href="#">FormFieldset::\$fieldset_display_required</a>	10
<a href="#">FormField::get_errors()</a>	9
<a href="#">FormField::add_errors()</a>	9
<a href="#">FormField::get_id()</a>	10
<a href="#">FormField::get_required_alert()</a>	10
<a href="#">FormFieldset</a>	10
<a href="#">FormField::throw_error()</a>	10
<a href="#">FormCheckbox::add_option()</a>	7
<a href="#">FormCheckbox::\$field_options</a>	7
<a href="#">FormBuilder::\$display_reset</a>	4
<a href="#">FormBuilder::\$display_preview</a>	4
<a href="#">FormBuilder::\$field_identifier_preview</a>	4
<a href="#">FormBuilder::\$form_action</a>	4
<a href="#">FormBuilder::\$form_fieldsets</a>	4
<a href="#">FormBuilder::\$form_class</a>	4
<a href="#">FormBuilder</a>	4
<a href="#">FIELD_TEXTAREA</a>	3
<a href="#">FIELD_INPUT_FILE</a>	3
<a href="#">FIELD_INPUT_CHECKBOX</a>	3
<a href="#">FIELD_INPUT_HIDDEN</a>	3
<a href="#">FIELD_INPUT_RADIO</a>	3
<a href="#">FIELD_SELECT</a>	3
<a href="#">FIELD_INPUT_TEXT</a>	3
<a href="#">FormBuilder::\$form_name</a>	4
<a href="#">FormBuilder::\$form_submit</a>	5
<a href="#">FormBuilder::set_form_action()</a>	6
<a href="#">FormBuilder::get_form_submit()</a>	6
<a href="#">FormBuilder::set_form_class()</a>	6
<a href="#">FormBuilder::set_form_name()</a>	6
<a href="#">FormCheckbox</a>	6
<a href="#">FormBuilder::set_form_submit()</a>	6
<a href="#">FormBuilder::get_form_name()</a>	6
<a href="#">FormBuilder::get_form_class()</a>	6
<a href="#">FormBuilder::display()</a>	5
<a href="#">FormBuilder::add_fieldset()</a>	5
<a href="#">FormBuilder::display_preview_button()</a>	5
<a href="#">FormBuilder::display_reset()</a>	5
<a href="#">FormBuilder::get_form_action()</a>	6

<a href="#">FormFieldset::get_fields()</a>	11
<a href="#">FormFieldset::get_title()</a>	11
<a href="#">Feed::\$data</a>	77
<a href="#">Feed</a>	77
<a href="#">Feed::\$id_cat</a>	77
<a href="#">Feed::\$module_id</a>	77
<a href="#">Feed::\$str</a>	77
<a href="#">Feed::\$name</a>	77
<a href="#">FEED_DATA_CLASS</a>	75
<a href="#">feed_data.class.php</a>	75
<a href="#">FormTextEdit::\$field_maxlength</a>	18
<a href="#">FormTextEdit</a>	17
<a href="#">FormTextEdit::\$field_size</a>	18
<a href="#">FormTextEdit::display()</a>	18
<a href="#">FEEDS_PATH</a>	74
<a href="#">feed.class.php</a>	74
<a href="#">Feed::\$tpl</a>	77
<a href="#">Feed::cache()</a>	78
<a href="#">Feed::read()</a>	80
<a href="#">Feed::load_file()</a>	79
<a href="#">Feed::update_cache()</a>	80
<a href="#">FeedData</a>	80
<a href="#">FeedData::\$date</a>	81
<a href="#">FeedData::\$auth_bit</a>	81
<a href="#">Feed::load_data()</a>	79
<a href="#">Feed::is_in_cache()</a>	79
<a href="#">Feed::export()</a>	78
<a href="#">Feed::clear_cache()</a>	78
<a href="#">Feed::get_cache_file_name()</a>	78
<a href="#">Feed::get_feed_menu()</a>	78
<a href="#">Feed::get_parsed()</a>	79
<a href="#">FormTextarea::display()</a>	17
<a href="#">FormTextarea::\$field_rows</a>	17
<a href="#">FormRadioChoice::\$field_options</a>	13
<a href="#">FormRadioChoice</a>	13
<a href="#">FormRadioChoice::add_option()</a>	14
<a href="#">FormRadioChoice::display()</a>	14
<a href="#">FormRadioChoiceOption::\$option_checked</a>	14
<a href="#">FormRadioChoiceOption</a>	14
<a href="#">FormHiddenField::display()</a>	13
<a href="#">FormHiddenField</a>	13
<a href="#">FormFieldset::set_title()</a>	11
<a href="#">FormFieldset::set_display_required()</a>	11
<a href="#">FormFieldset::throw_error()</a>	12
<a href="#">FormFileUploader</a>	12
<a href="#">FormFileUploader::display()</a>	13
<a href="#">FormFileUploader::\$field_size</a>	12
<a href="#">FormRadioChoiceOption::\$option_title</a>	15
<a href="#">FormRadioChoiceOption::display()</a>	15
<a href="#">FormSelectOption::display()</a>	16
<a href="#">FormSelectOption::\$option_title</a>	16
<a href="#">FormTextarea</a>	16
<a href="#">FormTextarea::\$field_cols</a>	17

<a href="#">FormTextarea::\$field forbidden tags</a>	17
<a href="#">FormTextarea::\$field editor</a>	17
<a href="#">FormSelectOption::\$option selected</a>	16
<a href="#">FormSelectOption</a>	16
<a href="#">FormSelect::\$field multiple</a>	15
<a href="#">FormSelect</a>	15
<a href="#">FormSelect::\$field options</a>	15
<a href="#">FormSelect::add_option()</a>	15
<a href="#">FormSelect::display()</a>	16
<a href="#">form_builder.class.php</a>	2

## G

<a href="#">get_module_name()</a>	266
<a href="#">get_ini_config()</a>	266
<a href="#">get_server_url_page()</a>	242
<a href="#">GENERATE CACHE AFTER THE OPERATION</a>	229
<a href="#">get_start_page()</a>	267
<a href="#">get_uid()</a>	267
<a href="#">gmdate format()</a>	267
<a href="#">get_utheme()</a>	267
<a href="#">get_ulang()</a>	267
<a href="#">Group::remove_member()</a>	191
<a href="#">Group::get_groups_array()</a>	191
<a href="#">GROUP TYPE</a>	186
<a href="#">GROUP DISABLE SELECT</a>	182
<a href="#">GROUP DISABLED ADVANCED AUTH</a>	182
<a href="#">GROUP DEFAULT IDSELECT</a>	182
<a href="#">Group</a>	190
<a href="#">Group::\$groups_auth</a>	190
<a href="#">Group::edit_member()</a>	190
<a href="#">Group::add_member()</a>	190
<a href="#">Group::\$groups_name</a>	190
<a href="#">groups.class.php</a>	182

## H

<a href="#">HTML_NO_PROTECT</a>	268
<a href="#">HTML_PROTECT</a>	268
<a href="#">html_entity_decode()</a>	267
<a href="#">htmlspecialchars_decode()</a>	267
<a href="#">HORIZONTAL SCROLLING MENU</a>	215
<a href="#">HORIZONTAL MENU</a>	215

## I

<a href="#">inc()</a>	268
<a href="#">INC_IMPORT</a>	268
<a href="#">import()</a>	268
<a href="#">images_stats.class.php</a>	240

<a href="#">INCORRECT DISPLAYING CONFIGURATION</a>	20
<a href="#">INTEGRATED IN ENVIRONMENT</a>	21
<a href="#">IGNORE AND CONTINUE BROWSING IF A CATEGORY DOES NOT MATCH</a>	20

## K

<a href="#">KERNEL_SCRIPT</a>	21
-------------------------------	----

## L

<a href="#">LinksMenuElement::get_image()</a>	222
<a href="#">LinksMenuElement::display()</a>	221
<i>Displays the menu according to the given template</i>	
<a href="#">LinksMenuElement::get_uid()</a>	222
<i>Returns the menu uid</i>	
<a href="#">LinksMenuElement::get_url()</a>	222
<a href="#">LinksMenuElement::set_image()</a>	222
<a href="#">LinksMenuElement::cache_export_end()</a>	221
<a href="#">LinksMenuElement::cache_export_begin()</a>	221
<a href="#">LinksMenuElement::\$image</a>	220
<a href="#">LinksMenuElement::\$uid</a>	221
<a href="#">LinksMenuElement::\$url</a>	221
<a href="#">LinksMenuElement::cache_export()</a>	221
<a href="#">LinksMenuElement::set_url()</a>	222
<a href="#">LinksMenuElement::update_uid()</a>	223
<i>Update the menu uid</i>	
<a href="#">LIB_IMPORT</a>	268
<a href="#">load_ini_file()</a>	268
<a href="#">load_menu_lang()</a>	268
<a href="#">load_module_lang()</a>	269
<a href="#">LINK_START_PAGE</a>	241
<a href="#">LinksMenuLink::display()</a>	224
<a href="#">LinksMenuElement::assign()</a>	223
<a href="#">LinksMenuElement::parent()</a>	223
<a href="#">LinksMenuLink</a>	223
<a href="#">LinksMenuLink::cache_export()</a>	224
<a href="#">LinksMenuElement::\$depth</a>	220
<a href="#">LinksMenuElement</a>	220
<a href="#">LINKS_MENU_ELEMENT_FULL_DISPLAYING</a>	216
<a href="#">LINKS_MENU_ELEMENT_CLASSIC_DISPLAYING</a>	216
<a href="#">links_menu_link.class.php</a>	217
<a href="#">LINKS_MENU_LINK_CLASS</a>	217
<a href="#">LinksMenu</a>	217
<a href="#">LINKS_MENU_ELEMENT_CLASS</a>	216
<a href="#">links_menu_element.class.php</a>	216
<a href="#">LOW_PRIORITY</a>	115
<a href="#">LOCK</a>	171
<a href="#">links_menu.class.php</a>	215
<a href="#">LINKS_MENU_CLASS</a>	215
<a href="#">LinksMenu::\$elements</a>	217
<a href="#">LinksMenu::\$type</a>	217

<a href="#">LinksMenu::get_type()</a>	219
<a href="#">LinksMenu::set_type()</a>	219
<i>Sets the type of the menu</i>	
<a href="#">LinksMenu::update_uid()</a>	219
<i>Update the menu uid</i>	
<a href="#">LinksMenu::parent()</a>	219
<a href="#">LinksMenu::get_menu_types_list()</a>	219
<i>static method which returns all the menu types</i>	
<a href="#">LinksMenu::get_children()</a>	219
<a href="#">LinksMenu::add()</a>	218
<a href="#">LinksMenu::add_array()</a>	218
<a href="#">LinksMenu::cache_export()</a>	218
<a href="#">LinksMenu::display()</a>	218
<a href="#">LOAD CACHE</a>	20

## M

<a href="#">ModuleMiniMenu::\$filename</a>	226
<a href="#">ModuleMiniMenu::cache_export()</a>	226
<a href="#">ModuleMiniMenu::get_title()</a>	226
<a href="#">ModuleMiniMenu</a>	226
<a href="#">MODULE MINI MENU CLASS</a>	226
<a href="#">MiniMenu::\$function name</a>	225
<a href="#">MiniMenu::cache_export()</a>	225
<a href="#">module_mini_menu.class.php</a>	226
<a href="#">module_interface.class.php</a>	228
<a href="#">MODULE ATTRIBUTE DOES NOT EXIST</a>	228
<a href="#">MODULE INSTALLED</a>	229
<a href="#">MODULE UNINSTALLED</a>	229
<a href="#">ModuleInterface</a>	230
<a href="#">MODULE FILES COULD NOT BE DROPPED</a>	229
<a href="#">MODULE ALREADY INSTALLED</a>	229
<a href="#">MODULE NOT AVAILABLE</a>	228
<i>module_interface.</i>	

```

-----
begin      : January 15, 2008
copyright  : (C) 2008 Loïc Rouchon
email      : horn@phpboost.

```

```
#####
```

*This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.*

<a href="#">MODULE NOT YET IMPLEMENTED</a>	228
<a href="#">MiniMenu</a>	225
<a href="#">MINI MENU CLASS</a>	225
<a href="#">Menu::get_block()</a>	207
<a href="#">Menu::get_block_position()</a>	207
<a href="#">Menu::get_id()</a>	207

<a href="#">Menu::get_auth()</a>	207
<a href="#">Menu::enabled()</a>	207
<a href="#">Menu::cache_export_begin()</a>	207
<a href="#">Menu::cache_export_end()</a>	207
<a href="#">Menu::display()</a>	207
<a href="#">Menu::get_title()</a>	207
<a href="#">Menu::id()</a>	208
<a href="#">Menu::set_title()</a>	208
<a href="#">Menu::assign()</a>	208
<a href="#">mini_menu.class.php</a>	225
<a href="#">Menu::set_block_position()</a>	208
<a href="#">Menu::set_block()</a>	208
<a href="#">Menu::is_enabled()</a>	208
<a href="#">Menu::set_auth()</a>	208
<a href="#">ModuleInterface::\$attributes</a>	230
<a href="#">ModuleInterface::\$enabled</a>	230
<a href="#">ModulesDiscoveryService::get_module()</a>	235
<a href="#">MiniCalendar</a>	251
<a href="#">MiniCalendar::\$date</a>	251
<a href="#">ModulesDiscoveryService::get_available_modules()</a>	234
<a href="#">ModulesDiscoveryService::get_all_modules()</a>	234
<a href="#">ModulesDiscoveryService::\$available_modules</a>	234
<a href="#">ModulesDiscoveryService::\$loaded_modules</a>	234
<a href="#">ModulesDiscoveryService::functionality()</a>	234
<a href="#">MiniCalendar::\$form_name</a>	251
<a href="#">MiniCalendar::\$num_instance</a>	252
<a href="#">MiniCalendar::set_date()</a>	253
<a href="#">MiniCalendar::set_style()</a>	253
<a href="#">MAGIC_QUOTES_DISABLED</a>	269
<a href="#">MiniCalendar::retrieve_date()</a>	252
<a href="#">MiniCalendar::get_date()</a>	252
<a href="#">MiniCalendar::\$style</a>	252
<a href="#">MiniCalendar::display()</a>	252
<a href="#">ModulesDiscoveryService</a>	234
<a href="#">ModuleInterface::unset_attribute()</a>	233
<a href="#">ModuleInterface::functionality()</a>	232
<a href="#">ModuleInterface::get_attribute()</a>	232
<a href="#">ModuleInterface::get_errors()</a>	232
<a href="#">ModuleInterface::\$name</a>	231
<a href="#">ModuleInterface::\$infos</a>	231
<a href="#">ModuleInterface::\$errors</a>	230
<a href="#">ModuleInterface::\$functionalities</a>	231
<a href="#">ModuleInterface::\$id</a>	231
<a href="#">ModuleInterface::get_id()</a>	232
<a href="#">ModuleInterface::get_infos()</a>	232
<a href="#">ModuleInterface::is_enabled()</a>	233
<a href="#">ModuleInterface::set_attribute()</a>	233
<a href="#">ModuleInterface::set_error()</a>	233
<a href="#">ModuleInterface::has_functionality()</a>	233
<a href="#">ModuleInterface::has_functionalities()</a>	232
<a href="#">ModuleInterface::get_name()</a>	232
<a href="#">ModuleInterface::got_error()</a>	232
<a href="#">Menu::cache_export()</a>	207

<a href="#">Menu::admin_display()</a>	207
<a href="#">MenuService::load()</a>	110
<a href="#">MenuService::move()</a>	110
<a href="#">MenuService::save()</a>	110
<a href="#">MenuService::get_menu_list()</a>	110
<a href="#">MenuService::get_menus_map()</a>	110
<a href="#">MenuService::enable()</a>	109
<a href="#">MenuService::enable_all()</a>	109
<a href="#">MenuService::generate_cache()</a>	109
<a href="#">MenuService::str_to_location()</a>	111
<a href="#">MenuService::update_mini_menus_list()</a>	111
<a href="#">MIME_FORMAT_HTML</a>	155
<a href="#">MIME_FORMAT_TEXT</a>	155
<a href="#">Mail</a>	158
<a href="#">mail.class.php</a>	155
<a href="#">mysql.class.php</a>	115
<a href="#">MenuService::update_mini_modules_list()</a>	111
<a href="#">MenuService::website_modules()</a>	111
<a href="#">MenuService::disable()</a>	109
<a href="#">MenuService::delete_module_feeds_menus()</a>	109
<a href="#">ModuleMap::set_description()</a>	59
<a href="#">menu_service.class.php</a>	94
<a href="#">MOVE_DOWN</a>	94
<a href="#">ModuleMap::get_description()</a>	59
<a href="#">ModuleMap::export()</a>	58
<a href="#">MOVE_CATEGORY_UP</a>	20
<a href="#">ModuleMap</a>	58
<a href="#">ModuleMap::\$description</a>	58
<a href="#">MOVE_UP</a>	94
<a href="#">MenuService</a>	107
<a href="#">MenuService::delete()</a>	108
<a href="#">MenuService::delete_mini_menu()</a>	108
<a href="#">MenuService::delete_mini_module()</a>	108
<a href="#">MenuService::change_position()</a>	108
<a href="#">MenuService::assign_positions_conditions()</a>	107
<a href="#">MenuService::add_mini_menu()</a>	107
<a href="#">MenuService::add_mini_module()</a>	107
<a href="#">Mail::\$content</a>	158
<a href="#">Mail::\$format</a>	158
<a href="#">MENU_AUTH_BIT</a>	204
<a href="#">MENU_ENABLED</a>	204
<a href="#">MENU_ENABLE_OR_NOT</a>	204
<a href="#">menu.class.php</a>	204
<a href="#">Mail::generate_headers()</a>	162
<a href="#">Mail::set_object()</a>	162
<a href="#">Mail::set_recipients()</a>	162
<a href="#">Mail::set_sender()</a>	162
<a href="#">MENU_NOT_ENABLED</a>	204
<a href="#">MENU_CLASS</a>	204
<a href="#">Menu::\$id</a>	206
<a href="#">Menu::\$position</a>	206
<a href="#">Menu::\$title</a>	206
<a href="#">Menu::\$enabled</a>	206

<a href="#">Menu::\$block</a>	205
<a href="#">Menu</a>	205
<a href="#">Menu::\$auth</a>	205
<a href="#">Mail::set_mime()</a>	162
<i>Sets the MIME type of the mail content</i>	
<a href="#">Mail::set_headers()</a>	161
<a href="#">Mail::\$sender_name</a>	159
<a href="#">Mail::check_validity()</a>	160
<a href="#">Mail::get_content()</a>	160
<a href="#">Mail::\$sender_mail</a>	159
<a href="#">Mail::\$recipients</a>	159
<a href="#">Mail::\$headers</a>	159
<a href="#">Mail::\$object</a>	159
<a href="#">Mail::get_headers()</a>	160
<a href="#">Mail::get_mime()</a>	160
<i>Returns the MIME type of the mail content</i>	
<a href="#">Mail::send()</a>	161
<a href="#">Mail::send_from_properties()</a>	161
<a href="#">Mail::set_content()</a>	161
<a href="#">Mail::get_sender_name()</a>	160
<a href="#">Mail::get_sender_mail()</a>	160
<a href="#">Mail::get_object()</a>	160
<a href="#">Mail::get_recipients()</a>	160
<a href="#">MOVE CATEGORY DOWN</a>	20

## N

<a href="#">NO UNIQ NAME</a>	157
<a href="#">NOTCLOSEFILE</a>	171
<a href="#">NOCHECK_PM_BOX</a>	183
<a href="#">NO_AUTOCONNECT</a>	184
<a href="#">NO_DELETE_ON_ERROR</a>	157
<a href="#">NO_LINE_ERROR</a>	93
<a href="#">NO_FATAL_ERROR_CACHE</a>	92
<a href="#">NO_ARCHIVE_ERROR</a>	93
<a href="#">NO_FILE_ERROR</a>	93
<a href="#">NOT_INSTALLED_MODULE</a>	229
<a href="#">NO_ALLOCATE_COLOR</a>	240
<a href="#">NO_FATAL_ERROR</a>	269
<a href="#">NO_UPDATE_PAGES</a>	269
<a href="#">number_round()</a>	269
<a href="#">numeric()</a>	269
<a href="#">NO_EDITOR_UNPARSE</a>	269
<a href="#">NO_PREVIOUS_NEXT_LINKS</a>	241
<a href="#">NO_DRAW_LEGEND</a>	240
<a href="#">NO_DRAW_PERCENT</a>	240
<a href="#">NO_DRAW_VALUES</a>	240
<a href="#">Note::get_attribute()</a>	36
<a href="#">Note::display_img()</a>	36
<a href="#">NOTE_DISPLAY_BLOCK</a>	22
<a href="#">NOTE_DISPLAY_NOTE</a>	22
<a href="#">NOTE_NODISPLAY_NBRNOTES</a>	22

<a href="#">Note</a>	34
<a href="#">note.class.php</a>	22
<a href="#">NOT RECURSIVE EXPLORATION</a>	20
<a href="#">NEW PARENT CATEGORY DOES NOT EXIST</a>	20
<a href="#">NEW STATUS UNKNOWN</a>	20
<a href="#">NORMAL MODE</a>	20
<a href="#">Note::\$idprov</a>	34
<a href="#">Note::\$module_folder</a>	34
<a href="#">Note::\$sql_table</a>	35
<a href="#">Note::add()</a>	35
<a href="#">Note::display_form()</a>	35
<a href="#">Note::\$script_path</a>	35
<a href="#">Note::\$script</a>	35
<a href="#">Note::\$notation_scale</a>	34
<a href="#">Note::\$options</a>	35
<a href="#">Note::\$path</a>	35
<a href="#">NEW CATEGORY IS IN ITS CHILDRENS</a>	20

## O

<a href="#">of_class()</a>	270
<a href="#">OPEN NOW</a>	172
<a href="#">OPEN AFTER</a>	172

## P

<a href="#">PackagesManager</a>	235
<a href="#">PackagesManager::install_module()</a>	235
<a href="#">PackagesManager::uninstall_module()</a>	236
<a href="#">pagination.class.php</a>	241
<a href="#">PHP VERSION CONFLICT</a>	229
<a href="#">packages_manager.class.php</a>	229
<a href="#">PrivateMsg::delete()</a>	192
<a href="#">PrivateMsg::delete_conversation()</a>	192
<a href="#">PrivateMsg::send()</a>	192
<a href="#">PrivateMsg::start_conversation()</a>	193
<a href="#">Pagination</a>	253
<a href="#">Pagination::\$nbr_end_links</a>	254
<a href="#">Pagination::get_first_msg()</a>	254
<a href="#">pages_displayed()</a>	270
<a href="#">parse_ini_array()</a>	270
<a href="#">phpboost version()</a>	270
<a href="#">Pagination::get_current_page()</a>	254
<a href="#">Pagination::display()</a>	254
<a href="#">Pagination::\$nbr_start_links</a>	254
<a href="#">Pagination::\$page</a>	254
<a href="#">Pagination::\$var_page</a>	254
<a href="#">PrivateMsg::count_conversations()</a>	191
<a href="#">PrivateMsg::\$pm_msg_id</a>	191
<a href="#">Parser::\$array_tags</a>	52
<a href="#">Parser::\$content</a>	53

<a href="#">Parser::\$page_path</a>	53
<a href="#">Parser::\$path_to_root</a>	53
<a href="#">Parser</a>	52
<a href="#">PICK UP</a>	44
<a href="#">POP UP WINDOW</a>	21
<a href="#">parser.class.php</a>	44
<a href="#">PARSER DO NOT STRIP SLASHES</a>	44
<a href="#">PARSER STRIP SLASHES</a>	44
<a href="#">Parser::get_content()</a>	53
<a href="#">Parser::get_page_path()</a>	53
<i>Returns the page path</i>	
<a href="#">PHP MIN VERSION UPDATES</a>	96
<a href="#">pm.class.php</a>	183
<a href="#">PrivateMsg</a>	191
<a href="#">PrivateMsg::\$pm_convers_id</a>	191
<a href="#">PHPBOOST OFFICIAL REPOSITORY</a>	96
<a href="#">Parser::set_path_to_root()</a>	54
<i>Sets the reference path for relative URL</i>	
<a href="#">Parser::get_path_to_root()</a>	54
<i>Returns the path to root attribute.</i>	
<a href="#">Parser::set_content()</a>	54
<a href="#">Parser::set_page_path()</a>	54
<i>Sets the page path</i>	
<a href="#">PRODUCTION MODE</a>	20

## R

<a href="#">READ_WRITE</a>	171
<a href="#">READ</a>	171
<a href="#">Repository::get_url()</a>	112
<a href="#">RANK_TYPE</a>	186
<a href="#">redirect()</a>	270
<a href="#">retrieve()</a>	271
<a href="#">req()</a>	271
<a href="#">redirect_confirm()</a>	271
<a href="#">Repository::check()</a>	112
<a href="#">Repository::\$xml</a>	112
<a href="#">RSS</a>	89
<a href="#">rss.class.php</a>	76
<a href="#">REIMPLANT</a>	44
<a href="#">RSS::load_file()</a>	89
<a href="#">RELOAD_CACHE</a>	92
<a href="#">Repository::\$url</a>	112
<a href="#">Repository</a>	111
<a href="#">RECURSIVE EXPLORATION</a>	20

## S

<a href="#">Sql::highlight_query()</a>	123
<a href="#">Sql::indent_query()</a>	124
<a href="#">Sql::insert_id()</a>	124

<a href="#">Sql::limit()</a>	124
<a href="#">Sql::get_executed_requests_number()</a>	123
<a href="#">Sql::get_dbms_version()</a>	123
<a href="#">Sql::fetch_assoc()</a>	122
<a href="#">Sql::fetch_row()</a>	123
<a href="#">Sql::get_data_base_name()</a>	123
<a href="#">Sql::list_databases()</a>	124
<a href="#">Sql::list_fields()</a>	124
<a href="#">Sql::query_array()</a>	126
<a href="#">Sql::query_close()</a>	126
<a href="#">Sql::query_inject()</a>	126
<a href="#">Sql::query_while()</a>	127
<a href="#">Sql::query()</a>	125
<a href="#">Sql::parse()</a>	125
<a href="#">Sql::list_tables()</a>	125
<a href="#">Sql::num_rows()</a>	125
<a href="#">Sql::optimize_tables()</a>	125
<a href="#">Sql::escape()</a>	122
<a href="#">Sql::drop_tables()</a>	122
<a href="#">StatsSaver::compute_users()</a>	113
<a href="#">Sql</a>	118
<a href="#">Sql::\$base_name</a>	119
<a href="#">Sql::\$connected</a>	119
<a href="#">StatsSaver::compute_referer()</a>	113
<a href="#">StatsSaver</a>	112
<a href="#">SiteMapSection::set_depth()</a>	71
<a href="#">SiteMapSection::set_link()</a>	72
<a href="#">stats_saver.class.php</a>	95
<a href="#">Sql::\$link</a>	119
<a href="#">Sql::\$req</a>	119
<a href="#">Sql::connect()</a>	121
<a href="#">Sql::count_table()</a>	121
<a href="#">Sql::create_database()</a>	122
<a href="#">Sql::date_diff()</a>	122
<a href="#">Sql::concat()</a>	121
<a href="#">Sql::close()</a>	120
<a href="#">Sql::affected_rows()</a>	120
<a href="#">Sql::auto_connect()</a>	120
<a href="#">Sql::clean_database_name()</a>	120
<i>Cleans the data base name to be sure it's a correct name</i>	
<a href="#">Sql::repair_tables()</a>	127
<a href="#">Sql::truncate_tables()</a>	127
<a href="#">Stats::\$nbr_color</a>	256
<a href="#">Stats::\$nbr_entry</a>	256
<a href="#">Stats::draw_ellipse()</a>	256
<a href="#">Stats::draw_graph()</a>	256
<a href="#">Stats::\$decimal</a>	256
<a href="#">Stats::\$data_stats</a>	255
<a href="#">Stats::\$array_allocated_color</a>	255
<a href="#">Stats::\$array_color_stats</a>	255
<a href="#">Stats::\$color_index</a>	255
<a href="#">Stats::draw_histogram()</a>	256
<a href="#">Stats::load_data()</a>	256

<a href="#">strprotect()</a>	274
<a href="#">strtodate()</a>	274
<a href="#">strtotimestamp()</a>	275
<a href="#">substr_html()</a>	275
<a href="#">strparse()</a>	273
<a href="#">strhash()</a>	273
<a href="#">second_parse()</a>	272
<a href="#">second_parse_url()</a>	272
<a href="#">set_subregex_multiplicity()</a>	273
<a href="#">Stats</a>	255
<a href="#">SERVER_URL</a>	243
<a href="#">Session::\$data</a>	193
<a href="#">Session::\$session_mod</a>	193
<a href="#">Session::act()</a>	193
<a href="#">Session::check()</a>	193
<a href="#">Session::\$autoconnect</a>	193
<a href="#">Session</a>	193
<a href="#">SYSTEM_PM</a>	183
<a href="#">session.class.php</a>	184
<a href="#">SEASURF_ATTACK_ERROR_PAGE</a>	184
<a href="#">Session::csrf_get_protect()</a>	194
<a href="#">Session::csrf_post_protect()</a>	194
<a href="#">Session::set_module_parameters()</a>	195
<a href="#">Session::start()</a>	195
<a href="#">securit_register_globals()</a>	242
<a href="#">Session::load()</a>	195
<a href="#">Session::get_token()</a>	195
<a href="#">Session::end()</a>	194
<a href="#">Session::garbage_collector()</a>	194
<a href="#">Session::get_module_parameters()</a>	195
<a href="#">SiteMapSection::get_link()</a>	71
<a href="#">SiteMapSection::export()</a>	71
<a href="#">SITE_MAP_PRIORITY_AVERAGE</a>	57
<a href="#">SITE_MAP_PRIORITY_HIGH</a>	57
<a href="#">SITE_MAP_PRIORITY_LOW</a>	57
<a href="#">SITE_MAP_PRIORITY_MAX</a>	57
<a href="#">SITE_MAP_FREQ_YEARLY</a>	57
<a href="#">SITE_MAP_FREQ_WEEKLY</a>	57
<a href="#">SITE_MAP_FREQ_HOURLY</a>	57
<a href="#">SITE_MAP_FREQ_MONTHLY</a>	57
<a href="#">SITE_MAP_FREQ_NEVER</a>	57
<a href="#">SITE_MAP_PRIORITY_MIN</a>	57
<a href="#">SITE_MAP_SEARCH_ENGINE_MODE</a>	57
<i>It will be for the search engines (sitemap. can be forgotten in that case.</i>	
<a href="#">SiteMap::build_kernel_map()</a>	60
<a href="#">SiteMap::build_modules_maps()</a>	61
<a href="#">SiteMap::export()</a>	61
<a href="#">SiteMap::get_site_name()</a>	61
<a href="#">SiteMap::add()</a>	60
<a href="#">SiteMap::\$site_name</a>	60
<a href="#">SITE_MAP_USER_MODE</a>	57
<i>It will be a page of the site containing the site map</i>	

<a href="#">SiteMap</a>	59
<a href="#">SiteMap::\$elements</a>	60
<a href="#">SITE MAP FREQ DEFAULT</a>	57
<a href="#">SITE MAP FREQ DAILY</a>	57
<a href="#">Search::\$modules</a>	37
<a href="#">Search::\$modules_conditions</a>	37
<a href="#">Search::\$search</a>	37
<a href="#">Search::get_ids()</a>	37
<a href="#">Search::\$id_user</a>	37
<a href="#">Search::\$id_search</a>	37
<a href="#">search.class.php</a>	23
<a href="#">Search</a>	36
<a href="#">Search::\$errors</a>	37
<a href="#">Search::get_results()</a>	37
<a href="#">Search::get_results_by_id()</a>	38
<a href="#">SITE MAP AUTH GUEST</a>	57
<i>The site map will be seen by every body, only the public elements must appear</i>	
<a href="#">SITE MAP AUTH USER</a>	57
<i>The site map is for the current user.</i>	
<a href="#">SITE MAP FREQ ALWAYS</a>	57
<a href="#">site_map.class.php</a>	57
<a href="#">Search::modules_in_cache()</a>	39
<a href="#">Search::insert_results()</a>	38
<a href="#">Search::is_in_cache()</a>	38
<a href="#">Search::is_search_id_in_cache()</a>	38
<a href="#">SiteMap::set_site_name()</a>	61
<a href="#">SiteMapElement</a>	61
<a href="#">SiteMapLink::get_last_modification_date()</a>	68
<a href="#">SiteMapLink::get_link()</a>	68
<a href="#">SiteMapLink::get_name()</a>	68
<a href="#">SiteMapLink::get_priority()</a>	68
<a href="#">SiteMapLink::get_change_freq()</a>	68
<a href="#">SiteMapLink::export()</a>	67
<a href="#">SiteMapLink::\$link</a>	67
<a href="#">SiteMapLink::\$name</a>	67
<a href="#">SiteMapLink::\$priority</a>	67
<a href="#">SiteMapLink::get_url()</a>	68
<a href="#">SiteMapLink::set_change_freq()</a>	68
<a href="#">SiteMapSection</a>	70
<a href="#">SiteMapSection::\$elements</a>	70
<a href="#">SiteMapSection::\$link</a>	70
<a href="#">SiteMapSection::add()</a>	71
<a href="#">SiteMapLink::SiteMapLink()</a>	69
<a href="#">SiteMapLink::set_priority()</a>	69
<a href="#">SiteMapLink::set_last_modification_date()</a>	68
<a href="#">SiteMapLink::set_link()</a>	69
<a href="#">SiteMapLink::set_name()</a>	69
<a href="#">SiteMapLink::\$last_modification_date</a>	66
<a href="#">SiteMapLink::\$change_freq</a>	66
<a href="#">SitemapExportConfig</a>	63
<a href="#">SitemapExportConfig::\$link_file</a>	63
<a href="#">SitemapExportConfig::\$module_map_file</a>	64
<a href="#">SitemapExportConfig::\$section_file</a>	64

<a href="#">SiteMapElement::set_depth()</a>	63
<a href="#">SiteMapElement::get_name()</a>	63
<a href="#">SiteMapElement::\$depth</a>	62
<a href="#">SiteMapElement::export()</a>	62
<a href="#">SiteMapElement::get_depth()</a>	62
<a href="#">SitemapExportConfig::\$site_map_file</a>	64
<a href="#">SitemapExportConfig::get_link_stream()</a>	64
<a href="#">SitemapExportConfig::set_section_stream()</a>	65
<a href="#">SitemapExportConfig::set_site_map_stream()</a>	65
<a href="#">SiteMapLink</a>	66
<a href="#">SitemapExportConfig::set_module_map_stream()</a>	65
<a href="#">SitemapExportConfig::set_link_stream()</a>	65
<a href="#">SitemapExportConfig::get_module_map_stream()</a>	65
<a href="#">SitemapExportConfig::get_section_stream()</a>	65
<a href="#">SitemapExportConfig::get_site_map_stream()</a>	65
<a href="#">STOP BROWSING IF A CATEGORY DOES NOT MATCH</a>	20

## T

<a href="#">Template::get_module_data_path()</a>	166
<a href="#">Template::parse()</a>	166
<a href="#">Template::pparse()</a>	166
<a href="#">Template::set_filenames()</a>	167
<a href="#">Template::copy()</a>	166
<a href="#">Template::assign_vars()</a>	166
<a href="#">Template::\$return_mode</a>	164
<a href="#">Template::\$template</a>	165
<a href="#">Template::\$tpl</a>	165
<a href="#">Template::assign_block_vars()</a>	165
<a href="#">Template::unassign_block_vars()</a>	167
<a href="#">Template::check_cache_file()</a>	167
<a href="#">TIMEZONE SITE</a>	275
<a href="#">TIMEZONE SYSTEM</a>	275
<a href="#">TIMEZONE USER</a>	275
<a href="#">to_js_string()</a>	275
<a href="#">TIMEZONE_AUTO</a>	239
<a href="#">TREE MENU</a>	215
<a href="#">Template::check_file()</a>	168
<a href="#">Template::include()</a>	168
<a href="#">Template::load()</a>	168
<a href="#">Template::\$module_data_path</a>	164
<a href="#">Template::\$files</a>	164
<a href="#">TPL BRACES STYLE</a>	45
<a href="#">TPL KEYWORD STYLE</a>	45
<a href="#">TPL NESTED VARIABLE STYLE</a>	45
<a href="#">TPL SHARP STYLE</a>	45
<a href="#">template_highlighter.class.php</a>	45
<a href="#">TINYMCE LANGUAGE</a>	43
<a href="#">TinyMCEEditor::\$array_tags</a>	41
<a href="#">TinyMCEEditor::\$array_tags2</a>	41
<a href="#">TinyMCEEditor::\$array_tags3</a>	41
<a href="#">TinyMCEEditor::display()</a>	41

<a href="#">TPL_VARIABLE_STYLE</a>	45
<a href="#">TemplateHighlighter</a>	55
<a href="#">template.class.php</a>	156
<a href="#">TEMPLATE_STRING_MODE</a>	156
<a href="#">TEMPLATE_WRITE_MODE</a>	156
<a href="#">Template</a>	163
<a href="#">TinyMCEUnparser::parse()</a>	56
<a href="#">TinyMCEUnparser</a>	56
<a href="#">TemplateHighlighter::parse()</a>	55
<a href="#">TinyMCEParser</a>	55
<a href="#">TinyMCEParser::parse()</a>	56
<a href="#">TinyMCEEditor</a>	41

## U

<a href="#">unusual_functions.inc.php</a>	242
<a href="#">UNEXISTING_MODULE</a>	229
<a href="#">User::update_user_theme()</a>	201
<a href="#">url.class.php</a>	243
<a href="#">URL_CLASS</a>	243
<a href="#">Url::\$path_to_root</a>	257
<a href="#">Url::\$is_relative</a>	257
<a href="#">Url</a>	257
<a href="#">User::update_user_lang()</a>	201
<a href="#">User::set_user_theme()</a>	201
<a href="#">User::check_max_value()</a>	200
<a href="#">User::check_level()</a>	200
<a href="#">User::check_auth()</a>	199
<a href="#">User::get_attribute()</a>	200
<a href="#">User::get_groups()</a>	200
<a href="#">User::set_user_lang()</a>	201
<a href="#">User::get_id()</a>	201
<a href="#">User::get_group_color()</a>	200
<a href="#">Url::\$server_url</a>	257
<a href="#">Url::\$url</a>	257
<a href="#">Url::relative()</a>	262
<a href="#">Url::path_to_root()</a>	261
<a href="#">Url::is_relative()</a>	261
<a href="#">Url::root_to_local()</a>	262
<a href="#">Url::server_url()</a>	262
<a href="#">url_encode_rewrite()</a>	276
<a href="#">url()</a>	276
<a href="#">unparse()</a>	276
<a href="#">Url::html_convert_root_relative2relative()</a>	261
<a href="#">Url::html_convert_root_relative2absolute()</a>	260
<a href="#">Url::compress()</a>	259
<a href="#">Url::check_wellformness()</a>	258
<a href="#">Url::absolute()</a>	258
<a href="#">Url::get_absolute_root()</a>	259
<a href="#">Url::get_relative()</a>	259
<a href="#">Url::html_convert_absolute2root_relative()</a>	260
<a href="#">Url::get_wellformness_regex()</a>	259

<a href="#">User::\$user_groups</a>	199
<a href="#">User::\$user_data</a>	199
<a href="#">Upload::validate_img()</a>	170
<a href="#">Upload::file()</a>	170
<a href="#">Upload::\$filename</a>	169
<a href="#">UPDATE_MBR_PM</a>	183
<a href="#">uploads.class.php</a>	185
<a href="#">Uploads</a>	196
<a href="#">USER_TYPE</a>	186
<a href="#">user.class.php</a>	186
<a href="#">Upload::\$extension</a>	169
<a href="#">Upload::\$error</a>	169
<a href="#">Updates::\$repositories</a>	113
<a href="#">Updates::\$apps</a>	113
<a href="#">Updates</a>	113
<a href="#">UNEXISTING_DATABASE</a>	115
<a href="#">upload.class.php</a>	157
<a href="#">Upload::\$base_directory</a>	169
<a href="#">Upload</a>	169
<a href="#">UNIQ_NAME</a>	157
<a href="#">Uploads::\$base_directory</a>	196
<a href="#">Uploads::\$error</a>	196
<a href="#">Uploads::Move_file()</a>	198
<a href="#">Uploads::Member_memory_used()</a>	197
<a href="#">Uploads::get_url()</a>	197
<a href="#">Uploads::Move_folder()</a>	198
<a href="#">Uploads::Rename_file()</a>	198
<a href="#">User::\$groups_auth</a>	199
<a href="#">User</a>	199
<a href="#">Uploads::Rename_folder()</a>	198
<a href="#">Uploads::get_img_mimetype()</a>	197
<a href="#">Uploads::get_admin_url()</a>	197
<a href="#">Uploads::Add_folder()</a>	196
<a href="#">Uploads::\$filename</a>	196
<a href="#">Uploads::\$extension</a>	196
<a href="#">Uploads::Del_file()</a>	196
<a href="#">Uploads::Del_folder()</a>	196
<a href="#">Uploads::Find_subfolder()</a>	197
<a href="#">Uploads::Empty_folder_member()</a>	197
<a href="#">updates.class.php</a>	96

## V

<a href="#">VERTICAL_SCROLLING_MENU</a>	215
<a href="#">VERTICAL_MENU</a>	215

## W

<a href="#">wordwrap_html()</a>	277
<a href="#">WRITE</a>	171