

OWL-PHP

010

Generated by Doxygen 1.6.3

Tue Dec 7 07:52:51 2010

Contents

1	Module Index	1
1.1	Modules	1
2	Class Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Module Documentation	9
5.1	Presentation modules	9
5.2	Business Object modules	10
5.3	Storage Object modules	11
5.4	Library (codes, messages files etc.)	12
6	Class Documentation	13
6.1	OWL Class Reference	13
6.1.1	Detailed Description	14
6.1.2	Constructor & Destructor Documentation	14
6.1.2.1	__destruct	14
6.1.3	Member Function Documentation	15
6.1.3.1	check	15
6.1.3.2	get_severity	15
6.1.3.3	get_status	15
6.1.3.4	init	15
6.1.3.5	reset	16
6.1.3.6	reset_calltree	16

6.1.3.7	set_high_severity	16
6.1.3.8	set_status	16
6.1.3.9	signal	17
6.1.3.10	traceback	17
6.1.4	Member Data Documentation	17
6.1.4.1	\$pstatus	17
6.1.4.2	\$severity	17
6.1.4.3	\$status	18
6.2	BaseElement Class Reference	19
6.2.1	Detailed Description	21
6.2.2	Member Function Documentation	21
6.2.2.1	check	21
6.2.2.2	get_severity	21
6.2.2.3	get_status	22
6.2.2.4	getAttributes	22
6.2.2.5	getEvents	22
6.2.2.6	init	22
6.2.2.7	reset	22
6.2.2.8	set_high_severity	23
6.2.2.9	set_status	23
6.2.2.10	setClass	23
6.2.2.11	setId	23
6.2.2.12	signal	24
6.2.2.13	traceback	24
6.2.3	Member Data Documentation	24
6.2.3.1	\$class	24
6.2.3.2	\$id	24
6.2.3.3	\$severity	24
6.2.3.4	\$style	25
6.3	ConfigHandler Class Reference	26
6.3.1	Detailed Description	26
6.3.2	Member Function Documentation	26
6.3.2.1	convert	26
6.3.2.2	get	26
6.3.2.3	read_config	27
6.3.2.4	set	27

6.4	DataHandler Class Reference	28
6.4.1	Detailed Description	30
6.4.2	Member Function Documentation	30
6.4.2.1	check	30
6.4.2.2	DataHandler	31
6.4.2.3	db	31
6.4.2.4	db_status	31
6.4.2.5	escape_string	31
6.4.2.6	find_field	32
6.4.2.7	get	32
6.4.2.8	get_severity	32
6.4.2.9	get_status	33
6.4.2.10	init	33
6.4.2.11	prepare	33
6.4.2.12	reset	33
6.4.2.13	reset	33
6.4.2.14	set	34
6.4.2.15	set_high_severity	34
6.4.2.16	set_join	34
6.4.2.17	set_key	34
6.4.2.18	set_status	35
6.4.2.19	set tablename	35
6.4.2.20	signal	35
6.4.2.21	traceback	36
6.4.3	Member Data Documentation	36
6.4.3.1	\$owl_data	36
6.4.3.2	\$owl_database	36
6.4.3.3	\$owl_joins	36
6.4.3.4	\$owl_keys	36
6.4.3.5	\$owl_prepared	36
6.4.3.6	\$owl_tablename	37
6.4.3.7	\$severity	37
6.5	DbHandler Class Reference	38
6.5.1	Detailed Description	41
6.5.2	Constructor & Destructor Documentation	41
6.5.2.1	__construct	41

6.5.2.2	<code>__destruct</code>	41
6.5.3	Member Function Documentation	41
6.5.3.1	<code>__clone</code>	41
6.5.3.2	<code>check</code>	42
6.5.3.3	<code>close</code>	42
6.5.3.4	<code>connect</code>	42
6.5.3.5	<code>create</code>	42
6.5.3.6	<code>dbread</code>	42
6.5.3.7	<code>escape_string</code>	43
6.5.3.8	<code>expand_field</code>	43
6.5.3.9	<code>extract_tablelist</code>	43
6.5.3.10	<code>get_instance</code>	44
6.5.3.11	<code>get_severity</code>	44
6.5.3.12	<code>get_status</code>	44
6.5.3.13	<code>init</code>	44
6.5.3.14	<code>is_open</code>	44
6.5.3.15	<code>last_inserted_id</code>	45
6.5.3.16	<code>open</code>	45
6.5.3.17	<code>prepare_delete</code>	45
6.5.3.18	<code>prepare_insert</code>	45
6.5.3.19	<code>prepare_read</code>	46
6.5.3.20	<code>prepare_update</code>	46
6.5.3.21	<code>read</code>	46
6.5.3.22	<code>reset</code>	47
6.5.3.23	<code>set_high_severity</code>	47
6.5.3.24	<code>set_query</code>	47
6.5.3.25	<code>set_status</code>	47
6.5.3.26	<code>signal</code>	48
6.5.3.27	<code>table_exists</code>	48
6.5.3.28	<code>tablelist</code>	48
6.5.3.29	<code>tablename</code>	49
6.5.3.30	<code>traceback</code>	49
6.5.3.31	<code>unescape_string</code>	49
6.5.3.32	<code>update_list</code>	50
6.5.3.33	<code>where_clause</code>	50
6.5.3.34	<code>write</code>	50

6.5.4	Member Data Documentation	50
6.5.4.1	\$database	50
6.5.4.2	\$db_prefix	51
6.5.4.3	\$errno	51
6.5.4.4	\$error	51
6.5.4.5	\$id	51
6.5.4.6	\$instance	51
6.5.4.7	\$opened	51
6.5.4.8	\$query	51
6.5.4.9	\$rowcount	51
6.5.4.10	\$severity	51
6.6	Dispatcher Class Reference	52
6.6.1	Detailed Description	54
6.6.2	Constructor & Destructor Documentation	54
6.6.2.1	__construct	54
6.6.3	Member Function Documentation	54
6.6.3.1	__clone	54
6.6.3.2	check	55
6.6.3.3	dispatch	55
6.6.3.4	get_instance	55
6.6.3.5	get_severity	55
6.6.3.6	get_status	56
6.6.3.7	init	56
6.6.3.8	reset	56
6.6.3.9	set_high_severity	56
6.6.3.10	set_status	56
6.6.3.11	signal	57
6.6.3.12	traceback	57
6.6.4	Member Data Documentation	58
6.6.4.1	\$instance	58
6.6.4.2	\$severity	58
6.7	FileHandler Class Reference	59
6.7.1	Detailed Description	61
6.7.2	Constructor & Destructor Documentation	61
6.7.2.1	__construct	61
6.7.2.2	__destruct	62

6.7.3	Member Function Documentation	62
6.7.3.1	check	62
6.7.3.2	close	62
6.7.3.3	download	62
6.7.3.4	encode	62
6.7.3.5	get_severity	62
6.7.3.6	get_status	63
6.7.3.7	init	63
6.7.3.8	open	63
6.7.3.9	read_data	63
6.7.3.10	read_line	64
6.7.3.11	reset	64
6.7.3.12	set_high_severity	64
6.7.3.13	set_status	64
6.7.3.14	signal	65
6.7.3.15	traceback	65
6.7.4	Member Data Documentation	65
6.7.4.1	\$pointer	65
6.7.4.2	\$name	66
6.7.4.3	\$opened	66
6.7.4.4	\$original_name	66
6.7.4.5	\$severity	66
6.8	Form Class Reference	67
6.8.1	Detailed Description	69
6.8.2	Constructor & Destructor Documentation	69
6.8.2.1	__construct	69
6.8.3	Member Function Documentation	70
6.8.3.1	addField	70
6.8.3.2	check	70
6.8.3.3	closeForm	70
6.8.3.4	get_severity	71
6.8.3.5	get_status	71
6.8.3.6	getAttributes	71
6.8.3.7	getEvents	71
6.8.3.8	init	71
6.8.3.9	openForm	72

6.8.3.10	reset	72
6.8.3.11	set_high_severity	72
6.8.3.12	set_status	72
6.8.3.13	setClass	73
6.8.3.14	setFieldAttributes	73
6.8.3.15	setId	73
6.8.3.16	showField	73
6.8.3.17	signal	73
6.8.3.18	traceback	74
6.8.4	Member Data Documentation	74
6.8.4.1	\$class	74
6.8.4.2	\$dispatcher	74
6.8.4.3	\$fields	74
6.8.4.4	\$id	74
6.8.4.5	\$severity	75
6.8.4.6	\$style	75
6.9	FormField Class Reference	76
6.9.1	Detailed Description	78
6.9.2	Constructor & Destructor Documentation	78
6.9.2.1	__construct	78
6.9.3	Member Function Documentation	78
6.9.3.1	check	78
6.9.3.2	get_severity	79
6.9.3.3	get_status	79
6.9.3.4	getAttributes	79
6.9.3.5	getEvents	79
6.9.3.6	getGenericFieldAttributes	80
6.9.3.7	getType	80
6.9.3.8	init	80
6.9.3.9	reset	80
6.9.3.10	set_high_severity	80
6.9.3.11	set_status	81
6.9.3.12	setClass	81
6.9.3.13	setDisabled	81
6.9.3.14	setId	81
6.9.3.15	setName	82

6.9.3.16	set Readonly	82
6.9.3.17	set Value	82
6.9.3.18	signal	82
6.9.3.19	traceback	83
6.9.4	Member Data Documentation	83
6.9.4.1	\$class	83
6.9.4.2	\$disabled	83
6.9.4.3	\$id	83
6.9.4.4	\$name	83
6.9.4.5	\$readonly	83
6.9.4.6	\$severity	83
6.9.4.7	\$style	83
6.9.4.8	\$type	84
6.9.4.9	\$value	84
6.10	FormFieldButton Class Reference	85
6.10.1	Detailed Description	87
6.10.2	Constructor & Destructor Documentation	87
6.10.2.1	__construct	87
6.10.3	Member Function Documentation	88
6.10.3.1	check	88
6.10.3.2	get_severity	88
6.10.3.3	get_status	88
6.10.3.4	getAttributes	88
6.10.3.5	getEvents	89
6.10.3.6	getFieldCode	89
6.10.3.7	getGenericFieldAttributes	89
6.10.3.8	getType	89
6.10.3.9	init	89
6.10.3.10	reset	90
6.10.3.11	set_high_severity	90
6.10.3.12	set_status	90
6.10.3.13	setClass	90
6.10.3.14	setDisabled	91
6.10.3.15	setId	91
6.10.3.16	setName	91
6.10.3.17	setReadonly	91

6.10.3.18 setValue	91
6.10.3.19 signal	91
6.10.3.20 traceback	92
6.10.4 Member Data Documentation	92
6.10.4.1 \$alt	92
6.10.4.2 \$class	92
6.10.4.3 \$disabled	92
6.10.4.4 \$id	92
6.10.4.5 \$name	92
6.10.4.6 \$readonly	93
6.10.4.7 \$severity	93
6.10.4.8 \$src	93
6.10.4.9 \$style	93
6.10.4.10 \$type	93
6.10.4.11 \$value	93
6.11 FormFieldCheckbox Class Reference	94
6.11.1 Detailed Description	96
6.11.2 Constructor & Destructor Documentation	96
6.11.2.1 __construct	96
6.11.3 Member Function Documentation	97
6.11.3.1 check	97
6.11.3.2 get_severity	97
6.11.3.3 get_status	97
6.11.3.4 getAttributes	97
6.11.3.5 getEvents	98
6.11.3.6 getFieldCode	98
6.11.3.7 getGenericFieldAttributes	98
6.11.3.8 getType	98
6.11.3.9 init	98
6.11.3.10 reset	99
6.11.3.11 set_high_severity	99
6.11.3.12 set_status	99
6.11.3.13 setChecked	99
6.11.3.14 setClass	100
6.11.3.15 setDisabled	100
6.11.3.16 setId	100

6.11.3.17 setName	100
6.11.3.18 setReadonly	100
6.11.3.19 setValue	100
6.11.3.20 signal	101
6.11.3.21 traceback	101
6.11.4 Member Data Documentation	101
6.11.4.1 \$checked	101
6.11.4.2 \$class	101
6.11.4.3 \$disabled	101
6.11.4.4 \$id	102
6.11.4.5 \$name	102
6.11.4.6 \$readonly	102
6.11.4.7 \$severity	102
6.11.4.8 \$style	102
6.11.4.9 \$type	102
6.11.4.10 \$value	102
6.12 FormFieldFile Class Reference	103
6.12.1 Detailed Description	105
6.12.2 Constructor & Destructor Documentation	105
6.12.2.1 __construct	105
6.12.3 Member Function Documentation	105
6.12.3.1 check	105
6.12.3.2 get_severity	106
6.12.3.3 get_status	106
6.12.3.4 getAttributes	106
6.12.3.5 getEvents	106
6.12.3.6 getGenericFieldAttributes	107
6.12.3.7 getType	107
6.12.3.8 init	107
6.12.3.9 reset	107
6.12.3.10 set_high_severity	107
6.12.3.11 set_status	108
6.12.3.12 setClass	108
6.12.3.13 setDisabled	108
6.12.3.14 setId	108
6.12.3.15 setName	109

6.12.3.16 setReadonly	109
6.12.3.17 setValue	109
6.12.3.18 signal	109
6.12.3.19 traceback	110
6.12.4 Member Data Documentation	110
6.12.4.1 \$class	110
6.12.4.2 \$disabled	110
6.12.4.3 \$id	110
6.12.4.4 \$name	110
6.12.4.5 \$readonly	110
6.12.4.6 \$severity	110
6.12.4.7 \$style	110
6.12.4.8 \$type	111
6.12.4.9 \$value	111
6.13 FormFieldRadio Class Reference	112
6.13.1 Detailed Description	114
6.13.2 Constructor & Destructor Documentation	114
6.13.2.1 __construct	114
6.13.3 Member Function Documentation	115
6.13.3.1 check	115
6.13.3.2 get_severity	115
6.13.3.3 get_status	115
6.13.3.4 getAttributes	115
6.13.3.5 getEvents	116
6.13.3.6 getFieldCode	116
6.13.3.7 getGenericFieldAttributes	116
6.13.3.8 getType	116
6.13.3.9 init	116
6.13.3.10 reset	117
6.13.3.11 set_high_severity	117
6.13.3.12 set_status	117
6.13.3.13 setClass	117
6.13.3.14 setDisabled	118
6.13.3.15 setId	118
6.13.3.16 setName	118
6.13.3.17 setReadonly	118

6.13.3.18 setSelected	118
6.13.3.19 setValue	118
6.13.3.20 signal	119
6.13.3.21 traceback	119
6.13.4 Member Data Documentation	119
6.13.4.1 \$class	119
6.13.4.2 \$disabled	119
6.13.4.3 \$id	119
6.13.4.4 \$name	120
6.13.4.5 \$readonly	120
6.13.4.6 \$selected	120
6.13.4.7 \$severity	120
6.13.4.8 \$style	120
6.13.4.9 \$type	120
6.13.4.10 \$value	120
6.14 FormFieldSelect Class Reference	121
6.14.1 Detailed Description	123
6.14.2 Constructor & Destructor Documentation	124
6.14.2.1 __construct	124
6.14.3 Member Function Documentation	124
6.14.3.1 check	124
6.14.3.2 get_severity	124
6.14.3.3 get_status	124
6.14.3.4 getAttributes	125
6.14.3.5 getEvents	125
6.14.3.6 getFieldCode	125
6.14.3.7 getGenericFieldAttributes	125
6.14.3.8 getType	125
6.14.3.9 init	126
6.14.3.10 reset	126
6.14.3.11 set_high_severity	126
6.14.3.12 set_status	126
6.14.3.13 setClass	127
6.14.3.14 setDisabled	127
6.14.3.15 setId	127
6.14.3.16 setMultiple	127

6.14.3.17 setName	127
6.14.3.18 setReadonly	127
6.14.3.19 setSize	128
6.14.3.20 setValue	128
6.14.3.21 signal	128
6.14.3.22 traceback	128
6.14.4 Member Data Documentation	129
6.14.4.1 \$class	129
6.14.4.2 \$disabled	129
6.14.4.3 \$id	129
6.14.4.4 \$multiple	129
6.14.4.5 \$name	129
6.14.4.6 \$options	129
6.14.4.7 \$readonly	129
6.14.4.8 \$severity	129
6.14.4.9 \$size	130
6.14.4.10 \$style	130
6.14.4.11 \$type	130
6.14.4.12 \$value	130
6.14.4.13 DefaultOptionGroup	130
6.15 FormFieldText Class Reference	131
6.15.1 Detailed Description	133
6.15.2 Constructor & Destructor Documentation	133
6.15.2.1 __construct	133
6.15.3 Member Function Documentation	134
6.15.3.1 check	134
6.15.3.2 get_severity	134
6.15.3.3 get_status	134
6.15.3.4 getAttributes	135
6.15.3.5 getEvents	135
6.15.3.6 getFieldCode	135
6.15.3.7 getGenericFieldAttributes	135
6.15.3.8 getType	135
6.15.3.9 init	136
6.15.3.10 reset	136
6.15.3.11 set_high_severity	136

6.15.3.12 set_status	136
6.15.3.13 setClass	137
6.15.3.14 setDisabled	137
6.15.3.15 setId	137
6.15.3.16 setMaxsize	137
6.15.3.17 setName	137
6.15.3.18 setReadonly	138
6.15.3.19 setSize	138
6.15.3.20 setValue	138
6.15.3.21 signal	138
6.15.3.22 traceback	138
6.15.4 Member Data Documentation	139
6.15.4.1 \$class	139
6.15.4.2 \$disabled	139
6.15.4.3 \$id	139
6.15.4.4 \$maxsize	139
6.15.4.5 \$name	139
6.15.4.6 \$readonly	139
6.15.4.7 \$severity	139
6.15.4.8 \$size	139
6.15.4.9 \$style	140
6.15.4.10 \$type	140
6.15.4.11 \$value	140
6.16 FormFieldTextarea Class Reference	141
6.16.1 Detailed Description	143
6.16.2 Constructor & Destructor Documentation	143
6.16.2.1 __construct	143
6.16.3 Member Function Documentation	144
6.16.3.1 check	144
6.16.3.2 get_severity	144
6.16.3.3 get_status	144
6.16.3.4 getAttributes	144
6.16.3.5 getEvents	145
6.16.3.6 getFieldCode	145
6.16.3.7 getGenericFieldAttributes	145
6.16.3.8 getType	145

6.16.3.9 <code>init</code>	145
6.16.3.10 <code>reset</code>	146
6.16.3.11 <code>set_high_severity</code>	146
6.16.3.12 <code>set_status</code>	146
6.16.3.13 <code>setClass</code>	146
6.16.3.14 <code>setDisabled</code>	147
6.16.3.15 <code>setId</code>	147
6.16.3.16 <code>setName</code>	147
6.16.3.17 <code>setReadonly</code>	147
6.16.3.18 <code>setValue</code>	147
6.16.3.19 <code>signal</code>	147
6.16.3.20 <code>traceback</code>	148
6.16.4 Member Data Documentation	148
6.16.4.1 <code>\$class</code>	148
6.16.4.2 <code>\$cols</code>	148
6.16.4.3 <code>\$disabled</code>	148
6.16.4.4 <code>\$id</code>	148
6.16.4.5 <code>\$name</code>	148
6.16.4.6 <code>\$readonly</code>	149
6.16.4.7 <code>\$rows</code>	149
6.16.4.8 <code>\$severity</code>	149
6.16.4.9 <code>\$style</code>	149
6.16.4.10 <code>\$type</code>	149
6.16.4.11 <code>\$value</code>	149
6.17 FormHandler Class Reference	150
6.17.1 Detailed Description	152
6.17.2 Constructor & Destructor Documentation	152
6.17.2.1 <code>__construct</code>	152
6.17.3 Member Function Documentation	153
6.17.3.1 <code>check</code>	153
6.17.3.2 <code>cleanString</code>	153
6.17.3.3 <code>get</code>	153
6.17.3.4 <code>get_instance</code>	154
6.17.3.5 <code>get_severity</code>	154
6.17.3.6 <code>get_status</code>	154
6.17.3.7 <code>init</code>	154

6.17.3.8 parse_formdata	155
6.17.3.9 reset	155
6.17.3.10 set	155
6.17.3.11 set_high_severity	155
6.17.3.12 set_status	155
6.17.3.13 signal	156
6.17.3.14 traceback	156
6.17.4 Member Data Documentation	157
6.17.4.1 \$instance	157
6.17.4.2 \$owl_formvalues	157
6.17.4.3 \$severity	157
6.18 ImageHandler Class Reference	158
6.18.1 Detailed Description	160
6.18.2 Constructor & Destructor Documentation	160
6.18.2.1 __construct	160
6.18.3 Member Function Documentation	161
6.18.3.1 check	161
6.18.3.2 close	161
6.18.3.3 download	161
6.18.3.4 encode	161
6.18.3.5 get_severity	161
6.18.3.6 get_status	162
6.18.3.7 init	162
6.18.3.8 open	162
6.18.3.9 read_data	162
6.18.3.10 read_line	163
6.18.3.11 reset	163
6.18.3.12 set_high_severity	163
6.18.3.13 set_status	163
6.18.3.14 signal	164
6.18.3.15 traceback	164
6.18.4 Member Data Documentation	164
6.18.4.1 \$fpointer	164
6.18.4.2 \$name	165
6.18.4.3 \$opened	165
6.18.4.4 \$original_name	165

6.18.4.5 \$severity	165
6.19 LogHandler Class Reference	166
6.19.1 Detailed Description	168
6.19.2 Constructor & Destructor Documentation	168
6.19.2.1 __construct	168
6.19.2.2 __destruct	169
6.19.3 Member Function Documentation	169
6.19.3.1 __clone	169
6.19.3.2 backtrace	169
6.19.3.3 check	169
6.19.3.4 close_logfile	169
6.19.3.5 compose_message	170
6.19.3.6 get_instance	170
6.19.3.7 get_severity	170
6.19.3.8 get_status	170
6.19.3.9 init	171
6.19.3.10 log	171
6.19.3.11 open_logfile	171
6.19.3.12 reset	171
6.19.3.13 set_filename	171
6.19.3.14 set_high_severity	171
6.19.3.15 set_status	172
6.19.3.16 signal	172
6.19.3.17 traceback	172
6.19.3.18 write_logfile	173
6.19.4 Member Data Documentation	173
6.19.4.1 \$filename	173
6.19.4.2 \$fpointer	173
6.19.4.3 \$instance	173
6.19.4.4 \$opened	173
6.19.4.5 \$severity	173
6.20 OldOFMStuff Class Reference	174
6.20.1 Member Function Documentation	174
6.20.1.1 OFM_FileDetails	174
6.20.1.2 OFM_FileInfo	174
6.20.1.3 OFM_IsAscii	174

6.21 OWL Class Reference	175
6.21.1 Detailed Description	177
6.21.2 Constructor & Destructor Documentation	177
6.21.2.1 <code>__construct</code>	177
6.21.3 Member Function Documentation	177
6.21.3.1 <code>__clone</code>	177
6.21.3.2 <code>check</code>	177
6.21.3.3 <code>factory</code>	178
6.21.3.4 <code>get_instance</code>	178
6.21.3.5 <code>get_severity</code>	178
6.21.3.6 <code>get_status</code>	179
6.21.3.7 <code>init</code>	179
6.21.3.8 <code>reset</code>	179
6.21.3.9 <code>set_high_severity</code>	179
6.21.3.10 <code>set_status</code>	179
6.21.3.11 <code>signal</code>	180
6.21.3.12 <code>stat</code>	180
6.21.3.13 <code>traceback</code>	180
6.21.4 Member Data Documentation	181
6.21.4.1 <code>\$instance</code>	181
6.21.4.2 <code>\$severity</code>	181
6.22 OWLException Class Reference	182
6.22.1 Detailed Description	182
6.22.2 Constructor & Destructor Documentation	182
6.22.2.1 <code>__construct</code>	182
6.22.3 Member Function Documentation	183
6.22.3.1 <code>check_hide</code>	183
6.22.3.2 <code>get_trace</code>	183
6.22.3.3 <code>stack_dump</code>	183
6.22.3.4 <code>traceback</code>	183
6.22.4 Member Data Documentation	184
6.22.4.1 <code>\$caller</code>	184
6.22.4.2 <code>\$hidden_args</code>	184
6.22.4.3 <code>\$thrown_code</code>	184
6.23 OWLExceptionHandler Class Reference	185
6.23.1 Detailed Description	185

6.23.2 Member Function Documentation	185
6.23.2.1 handle_exception	185
6.23.2.2 log_exception	185
6.24 OWLloader Class Reference	186
6.24.1 Detailed Description	186
6.24.2 Member Function Documentation	186
6.24.2.1 _tryLoad	186
6.24.2.2 getClass	186
6.25 Register Class Reference	188
6.25.1 Detailed Description	188
6.25.2 Member Function Documentation	188
6.25.2.1 get_code	188
6.25.2.2 get_run_id	188
6.25.2.3 get_severity	189
6.25.2.4 get_severity_level	189
6.25.2.5 init	189
6.25.2.6 register_app	189
6.25.2.7 register_class	189
6.25.2.8 register_code	190
6.25.2.9 register_messages	190
6.25.2.10 register_severity	190
6.25.2.11 set_application	190
6.25.2.12 set_class	190
6.25.2.13 set_severity	191
6.26 SchemeHandler Class Reference	192
6.26.1 Detailed Description	194
6.26.2 Constructor & Destructor Documentation	195
6.26.2.1 __construct	195
6.26.2.2 __destruct	195
6.26.3 Member Function Documentation	195
6.26.3.1 __clone	195
6.26.3.2 __define_field	195
6.26.3.3 alter_scheme	195
6.26.3.4 alter_table	196
6.26.3.5 check	196
6.26.3.6 compare	196

6.26.3.7	create_scheme	196
6.26.3.8	create_table	197
6.26.3.9	define_index	197
6.26.3.10	define_scheme	197
6.26.3.11	get_instance	197
6.26.3.12	get_severity	198
6.26.3.13	get_status	198
6.26.3.14	get_table_columns	198
6.26.3.15	get_table_indexes	198
6.26.3.16	init	199
6.26.3.17	reset	199
6.26.3.18	scheme	199
6.26.3.19	set_high_severity	199
6.26.3.20	set_status	199
6.26.3.21	signal	200
6.26.3.22	table_description	200
6.26.3.23	traceback	201
6.26.3.24	validate_scheme	201
6.26.4	Member Data Documentation	201
6.26.4.1	\$db	201
6.26.4.2	\$id	201
6.26.4.3	\$instance	201
6.26.4.4	\$inuse	201
6.26.4.5	\$scheme	202
6.26.4.6	\$severity	202
6.26.4.7	\$table	202
6.27	Session Class Reference	203
6.27.1	Detailed Description	205
6.27.2	Constructor & Destructor Documentation	205
6.27.2.1	__construct	205
6.27.2.2	__destruct	205
6.27.3	Member Function Documentation	205
6.27.3.1	check	205
6.27.3.2	close	206
6.27.3.3	destroy	206
6.27.3.4	gc	206

6.27.3.5 <code>get_session_var</code>	206
6.27.3.6 <code>get_severity</code>	207
6.27.3.7 <code>get_status</code>	207
6.27.3.8 <code>init</code>	207
6.27.3.9 <code>open</code>	207
6.27.3.10 <code>read</code>	207
6.27.3.11 <code>reset</code>	208
6.27.3.12 <code>set_high_severity</code>	208
6.27.3.13 <code>set_session_var</code>	208
6.27.3.14 <code>set_status</code>	208
6.27.3.15 <code>signal</code>	209
6.27.3.16 <code>traceback</code>	209
6.27.3.17 <code>write</code>	209
6.27.4 Member Data Documentation	210
6.27.4.1 <code>\$dataset</code>	210
6.27.4.2 <code>\$severity</code>	210
6.28 SessionHandler Class Reference	211
6.28.1 Detailed Description	213
6.28.2 Constructor & Destructor Documentation	213
6.28.2.1 <code>__construct</code>	213
6.28.2.2 <code>__destruct</code>	213
6.28.3 Member Function Documentation	214
6.28.3.1 <code>check</code>	214
6.28.3.2 <code>close</code>	214
6.28.3.3 <code>destroy</code>	214
6.28.3.4 <code>gc</code>	214
6.28.3.5 <code>get_severity</code>	215
6.28.3.6 <code>get_status</code>	215
6.28.3.7 <code>init</code>	215
6.28.3.8 <code>open</code>	215
6.28.3.9 <code>read</code>	215
6.28.3.10 <code>reset</code>	216
6.28.3.11 <code>set_high_severity</code>	216
6.28.3.12 <code>set_status</code>	216
6.28.3.13 <code>signal</code>	217
6.28.3.14 <code>traceback</code>	217

6.28.3.15	write	217
6.28.4	Member Data Documentation	218
6.28.4.1	\$dataset	218
6.28.4.2	\$db	218
6.28.4.3	\$severity	218
6.29	StatusHandler Class Reference	219
6.29.1	Detailed Description	219
6.29.2	Constructor & Destructor Documentation	220
6.29.2.1	__construct	220
6.29.3	Member Function Documentation	220
6.29.3.1	__clone	220
6.29.3.2	get_code	220
6.29.3.3	get_instance	220
6.29.3.4	get_message	220
6.29.3.5	get_severity	221
6.29.3.6	reset	221
6.29.3.7	set_code	221
6.29.3.8	set_params	221
6.29.4	Member Data Documentation	221
6.29.4.1	\$code	221
6.29.4.2	\$instance	221
6.29.4.3	\$params	222
6.30	User Class Reference	223
6.30.1	Detailed Description	225
6.30.2	Constructor & Destructor Documentation	225
6.30.2.1	__construct	225
6.30.2.2	__destruct	225
6.30.3	Member Function Documentation	226
6.30.3.1	check	226
6.30.3.2	get_id	226
6.30.3.3	get_session_var	226
6.30.3.4	get_severity	226
6.30.3.5	get_status	227
6.30.3.6	get_username	227
6.30.3.7	init	227
6.30.3.8	login	227

6.30.3.9	logout	227
6.30.3.10	reset	228
6.30.3.11	set_high_severity	228
6.30.3.12	set_session_var	228
6.30.3.13	set_status	228
6.30.3.14	set_username	229
6.30.3.15	signal	229
6.30.3.16	traceback	229
6.30.4	Member Data Documentation	230
6.30.4.1	\$dataset	230
6.30.4.2	\$session	230
6.30.4.3	\$severity	230
6.30.4.4	\$user_data	230
6.31	UserHandler Class Reference	231
6.31.1	Detailed Description	233
6.31.2	Constructor & Destructor Documentation	233
6.31.2.1	__construct	233
6.31.2.2	__destruct	234
6.31.3	Member Function Documentation	234
6.31.3.1	check	234
6.31.3.2	get_severity	234
6.31.3.3	get_status	234
6.31.3.4	get_username	235
6.31.3.5	hash_password	235
6.31.3.6	init	235
6.31.3.7	login	235
6.31.3.8	logout	236
6.31.3.9	read_userdata	236
6.31.3.10	reset	236
6.31.3.11	set_high_severity	236
6.31.3.12	set_status	236
6.31.3.13	set_username	237
6.31.3.14	signal	237
6.31.3.15	traceback	237
6.31.4	Member Data Documentation	238
6.31.4.1	\$dataset	238

6.31.4.2	\$session	238
6.31.4.3	\$severity	238
6.31.4.4	\$user_data	238
7	File Documentation	239
7.1	/home/oscar/projects/owl-php/src/config.php File Reference	239
7.1.1	Detailed Description	239
7.1.2	Variable Documentation	239
7.1.2.1	\$GLOBALS	239
7.1.2.2	\$GLOBALS	239
7.1.2.3	\$GLOBALS	239
7.1.2.4	\$GLOBALS	239
7.2	/home/oscar/projects/owl-php/src/index.php File Reference	241
7.2.1	Detailed Description	241
7.2.2	Enumeration Type Documentation	241
7.2.2.1	OWL_ROOT	241
7.2.3	Variable Documentation	241
7.2.3.1	\$_form	241
7.2.3.2	\$_index	241
7.2.3.3	\$_scheme	242
7.2.3.4	\$_t	242
7.2.3.5	\$_table	242
7.2.3.6	\$_user	242
7.2.3.7	\$LoginForm	242
7.2.3.8	switch	242
7.3	/home/oscar/projects/owl-php/src/kernel/bo/class.dispatcher.php File Reference	243
7.3.1	Detailed Description	243
7.4	/home/oscar/projects/owl-php/src/kernel/bo/class.form.php File Reference	244
7.4.1	Detailed Description	244
7.5	/home/oscar/projects/owl-php/src/kernel/bo/class.owl.php File Reference	245
7.5.1	Detailed Description	245
7.6	/home/oscar/projects/owl-php/src/kernel/bo/class.session.php File Reference	246
7.6.1	Detailed Description	246
7.7	/home/oscar/projects/owl-php/src/kernel/bo/class.user.php File Reference	247
7.7.1	Detailed Description	247
7.8	/home/oscar/projects/owl-php/src/kernel/class._owl.php File Reference	248
7.8.1	Detailed Description	248

7.9 /home/oscar/projects/owl-php/src/kernel/so/class.confighandler.php File Reference	249
7.9.1 Detailed Description	249
7.10 /home/oscar/projects/owl-php/src/kernel/so/class.datahandler.php File Reference	250
7.10.1 Detailed Description	250
7.10.2 Enumeration Type Documentation	251
7.10.2.1 DATA_DELETE	251
7.10.2.2 DATA_READ	251
7.10.2.3 DATA_RESET_FULL	251
7.10.2.4 DATA_RESET_META	251
7.10.2.5 DATA_RESET_PREPARE	251
7.10.2.6 DATA_RESET_STATUS	251
7.10.2.7 DATA_UNPREPARED	251
7.10.2.8 DATA_UPDATE	251
7.10.2.9 DATA_WRITE	251
7.11 /home/oscar/projects/owl-php/src/kernel/so/class.dbhandler.php File Reference	252
7.11.1 Detailed Description	252
7.11.2 Enumeration Type Documentation	252
7.11.2.1 DBHANDLE_DATA	252
7.11.2.2 DBHANDLE_FIELDCOUNT	253
7.11.2.3 DBHANDLE_ROWCOUNT	253
7.11.2.4 DBHANDLE_SINGLEFIELD	253
7.11.2.5 DBHANDLE_SINGLEROW	253
7.11.2.6 DBHANDLE_TOTALFIELDCOUNT	253
7.12 /home/oscar/projects/owl-php/src/kernel/so/class.exceptionhandler.php File Reference	254
7.12.1 Detailed Description	254
7.13 /home/oscar/projects/owl-php/src/kernel/so/class.filehandler.php File Reference	255
7.13.1 Detailed Description	255
7.13.2 Enumeration Type Documentation	255
7.13.2.1 FILE_NOTRIM	255
7.13.2.2 FILE_TRIM_C	255
7.13.2.3 FILE_TRIM_L	256
7.13.2.4 FILE_TRIM_R	256
7.14 /home/oscar/projects/owl-php/src/kernel/so/class.formhandler.php File Reference	257
7.14.1 Detailed Description	257
7.14.2 Enumeration Type Documentation	257
7.14.2.1 FORMDATA_BASIC_HTML	257

7.14.2.2 FORMDATA_CUSTOM	258
7.14.2.3 FORMDATA_EXTENDED_HTML	258
7.14.2.4 FORMDATA_FULL_HTML	258
7.14.2.5 FORMDATA_HTML_CODE	258
7.14.2.6 FORMDATA_RAW	258
7.14.2.7 FORMDATA_STRICT	258
7.15 /home/oscar/projects/owl-php/src/kernel/so/class.imagehandler.php File Reference	259
7.15.1 Detailed Description	259
7.15.2 Enumeration Type Documentation	259
7.15.2.1 FILE_NOTRIM	259
7.15.2.2 FILE_TRIM_C	259
7.15.2.3 FILE_TRIM_L	259
7.15.2.4 FILE_TRIM_R	260
7.16 /home/oscar/projects/owl-php/src/kernel/so/class.loghandler.php File Reference .	261
7.16.1 Detailed Description	261
7.17 /home/oscar/projects/owl-php/src/kernel/so/class.register.php File Reference .	262
7.17.1 Detailed Description	262
7.17.2 Enumeration Type Documentation	262
7.17.2.1 OWL_APPLICATION_PATTERN	262
7.17.2.2 OWL_OBJECT_PATTERN	262
7.17.2.3 OWL_SEVERITY_PATTERN	262
7.17.2.4 OWL_STATUS_PATTERN	262
7.18 /home/oscar/projects/owl-php/src/kernel/so/class.schemehandler.php File Reference	263
7.18.1 Detailed Description	263
7.19 /home/oscar/projects/owl-php/src/kernel/so/class.sessionhandler.php File Reference	264
7.19.1 Detailed Description	264
7.19.2 Enumeration Type Documentation	264
7.19.2.1 SESSIONVAR_ARRAY	264
7.19.2.2 SESSIONVAR_DECR	264
7.19.2.3 SESSIONVAR_INCR	265
7.19.2.4 SESSIONVAR_SET	265
7.19.2.5 SESSIONVAR_UNSET	265
7.20 /home/oscar/projects/owl-php/src/kernel/so/class.statushandler.php File Reference	266
7.20.1 Detailed Description	266
7.21 /home/oscar/projects/owl-php/src/kernel/so/class.userhandler.php File Reference	267
7.21.1 Detailed Description	267

7.22 /home/oscar/projects/owl-php/src/kernel/ui/class.baseelement.php File Reference	268
7.22.1 Detailed Description	268
7.23 /home/oscar/projects/owl-php/src/kernel/ui/class.formfield.php File Reference	269
7.23.1 Detailed Description	269
7.24 /home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.button.php File Reference	270
7.24.1 Detailed Description	270
7.25 /home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.checkbox.php File Reference	271
7.25.1 Detailed Description	271
7.26 /home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.file.php File Reference	272
7.26.1 Detailed Description	272
7.27 /home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.radio.php File Reference	273
7.27.1 Detailed Description	273
7.28 /home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.select.php File Reference	274
7.28.1 Detailed Description	274
7.29 /home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.text.php File Reference	275
7.29.1 Detailed Description	275
7.30 /home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.textarea.php File Reference	276
7.30.1 Detailed Description	276
7.31 /home/oscar/projects/owl-php/src/lib/owl.debug.functions.php File Reference	277
7.31.1 Detailed Description	277
7.31.2 Function Documentation	277
7.31.2.1 DBG_dumpval	277
7.32 /home/oscar/projects/owl-php/src/lib/owl.helper.functions.php File Reference	278
7.32.1 Detailed Description	278
7.33 /home/oscar/projects/owl-php/src/lib/owl.messages.en-uk.php File Reference	279
7.33.1 Detailed Description	279
7.33.2 Variable Documentation	279
7.33.2.1 \$GLOBALS	279
7.34 /home/oscar/projects/owl-php/src/lib/owl.messages.php File Reference	280
7.34.1 Detailed Description	280
7.34.2 Variable Documentation	280
7.34.2.1 \$GLOBALS	280
7.35 /home/oscar/projects/owl-php/src/lib/owl.nodebug.functions.php File Reference	281
7.35.1 Detailed Description	281
7.35.2 Function Documentation	281
7.35.2.1 DBG_dumpval	281

7.36 /home/oscar/projects/owl-php/src/lib/owl.severitycodes.php File Reference	282
7.36.1 Detailed Description	282
7.36.2 Enumeration Type Documentation	282
7.36.2.1 OWL_BUG	282
7.36.2.2 OWL_CRITICAL	282
7.36.2.3 OWL_DEBUG	282
7.36.2.4 OWL_ERROR	282
7.36.2.5 OWL_FATAL	282
7.36.2.6 OWL_INFO	283
7.36.2.7 OWL_OK	283
7.36.2.8 OWL_SUCCESS	283
7.36.2.9 OWL_WARNING	283
7.37 /home/oscar/projects/owl-php/src/OWLloader.php File Reference	284
7.37.1 Detailed Description	284
7.37.2 Enumeration Type Documentation	285
7.37.2.1 OWL_BO_INC	285
7.37.2.2 OWL_INCLUDE	285
7.37.2.3 OWL_LIBRARY	285
7.37.2.4 OWL_SITE_TOP	285
7.37.2.5 OWL_SO_INC	285
7.37.2.6 OWL_UI_INC	285
7.37.3 Variable Documentation	285
7.37.3.1 \$GLOBALS	285
7.37.3.2 \$GLOBALS	285
7.37.3.3 \$GLOBALS	285
7.37.3.4 if	285
7.38 /home/oscar/projects/owl-php/src/OWLrundown.php File Reference	286
7.38.1 Detailed Description	286
7.38.2 Variable Documentation	286
7.38.2.1 \$_form	286

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Presentation modules	9
Business Object modules	10
Storage Object modules	11
Library (codes, messages files etc.)	12

Chapter 2

Class Index

2.1 Class Hierarchy

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

_OWL	13
BaseElement	19
Form	67
FormField	76
FormFieldButton	85
FormFieldCheckbox	94
FormFieldFile	103
FormFieldRadio	112
FormFieldSelect	121
FormFieldText	131
FormFieldTextarea	141
DataHandler	28
DbHandler	38
Dispatcher	52
FileHandler	59
ImageHandler	158
FormHandler	150
LogHandler	166
OWL	175
SchemeHandler	192
SessionHandler	211
Session	203
UserHandler	231
User	223
ConfigHandler	26
OldOFMStuff	174
OWLEception	182
OWLEceptionHandler	185
OWLloader	186
Register	188
StatusHandler	219

Chapter 3

Class Index

3.1 Class List

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

_OWL	13
BaseElement (DOM Element base class)	19
ConfigHandler (Configuration handler)	26
DataHandler (The OWL Data object)	28
DbHandler (Database handler)	38
Dispatcher (Dispatcher singleton)	52
FileHandler (File handler)	59
Form (Form Element class)	67
FormField (Formfield)	76
FormFieldButton (Formfield)	85
FormFieldCheckbox (Formfield)	94
FormFieldFile (Formfield)	103
FormFieldRadio (Formfield)	112
FormFieldSelect (Formfield)	121
FormFieldText (Formfield)	131
FormFieldTextarea (Formfield)	141
FormHandler (Formhandler)	150
ImageHandler (Image handler)	158
LogHandler (Log handler)	166
OldOFMStuff	174
OWL	175
OWLException (Exception handler)	182
OWLExceptionHandler (Default exception handler)	185
OWLloader (Class loader)	186
Register	188
SchemeHandler (Scheme handler)	192
Session (OWL-PHP session object)	203
SessionHandler (PHP session object)	211
StatusHandler (Status object)	219
User (OWL-PHP session object)	223
UserHandler (User object)	231

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

/home/oscar/projects/owl-php/src/config.php	239
/home/oscar/projects/owl-php/src/index.php	241
/home/oscar/projects/owl-php/src/OWLloader.php	284
/home/oscar/projects/owl-php/src/OWLrundown.php	286
/home/oscar/projects/owl-php/src/kernel/class._owl.php	248
/home/oscar/projects/owl-php/src/kernel/bo/class.dispatcher.php	243
/home/oscar/projects/owl-php/src/kernel/bo/class.form.php	244
/home/oscar/projects/owl-php/src/kernel/bo/class.owl.php	245
/home/oscar/projects/owl-php/src/kernel/bo/class.session.php	246
/home/oscar/projects/owl-php/src/kernel/bo/class.user.php	247
/home/oscar/projects/owl-php/src/kernel/so/class.confighandler.php	249
/home/oscar/projects/owl-php/src/kernel/so/class.datahandler.php	250
/home/oscar/projects/owl-php/src/kernel/so/class.dbhandler.php	252
/home/oscar/projects/owl-php/src/kernel/so/class.exceptionhandler.php	254
/home/oscar/projects/owl-php/src/kernel/so/class.filehandler.php	255
/home/oscar/projects/owl-php/src/kernel/so/class.formhandler.php	257
/home/oscar/projects/owl-php/src/kernel/so/class.imagehandler.php	259
/home/oscar/projects/owl-php/src/kernel/so/class.loghandler.php	261
/home/oscar/projects/owl-php/src/kernel/so/class.register.php	262
/home/oscar/projects/owl-php/src/kernel/so/class.schemehandler.php	263
/home/oscar/projects/owl-php/src/kernel/so/class.sessionhandler.php	264
/home/oscar/projects/owl-php/src/kernel/so/class.statushandler.php	266
/home/oscar/projects/owl-php/src/kernel/so/class.userhandler.php	267
/home/oscar/projects/owl-php/src/kernel/ui/class.baseelement.php	268
/home/oscar/projects/owl-php/src/kernel/ui/class.formfield.php	269
/home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.button.php	270
/home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.checkbox.php	271
/home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.file.php	272
/home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.radio.php	273
/home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.select.php	274
/home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.text.php	275
/home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.textarea.php	276
/home/oscar/projects/owl-php/src/lib/owl.debug.functions.php	277

/home/oscar/projects/owl-php/src/lib/ owl.helper.functions.php	278
/home/oscar/projects/owl-php/src/lib/ owl.messages.en-uk.php	279
/home/oscar/projects/owl-php/src/lib/ owl.messages.php	280
/home/oscar/projects/owl-php/src/lib/ owl.nodebug.functions.php	281
/home/oscar/projects/owl-php/src/lib/ owl.severitycodes.php	282

Chapter 5

Module Documentation

5.1 Presentation modules

Classes

- class [BaseElement](#)
DOM Element base class.
- class [FormField](#)
Formfield.
- class [FormFieldButton](#)
Formfield.
- class [FormFieldCheckbox](#)
Formfield.
- class [FormFieldFile](#)
Formfield.
- class [FormFieldRadio](#)
Formfield.
- class [FormFieldSelect](#)
Formfield.
- class [FormFieldText](#)
Formfield.
- class [FormFieldTextarea](#)
Formfield.

5.2 Business Object modules

Classes

- class [Dispatcher](#)
Dispatcher singleton.
- class [Form](#)
Form Element class.
- class [Session](#)
the OWL-PHP session object
- class [User](#)
the OWL-PHP session object

5.3 Storage Object modules

Classes

- class [ConfigHandler](#)
Configuration handler.
- class [DataHandler](#)
The OWL Data object.
- class [DbHandler](#)
Database handler.
- class [OWLEception](#)
Exception handler.
- class [OWLEExceptionHandler](#)
Default exception handler.
- class [FileHandler](#)
File handler.
- class [FormHandler](#)
Formhandler.
- class [ImageHandler](#)
Image handler.
- class [LogHandler](#)
Log handler.
- class [Register](#)
- class [SchemeHandler](#)
Scheme handler.
- class [SessionHandler](#)
the PHP session object
- class [StatusHandler](#)
Status object.
- class [UserHandler](#)
the user object
- class [OWLloader](#)
Class loader.

5.4 Library (codes, messages files etc.)

Files

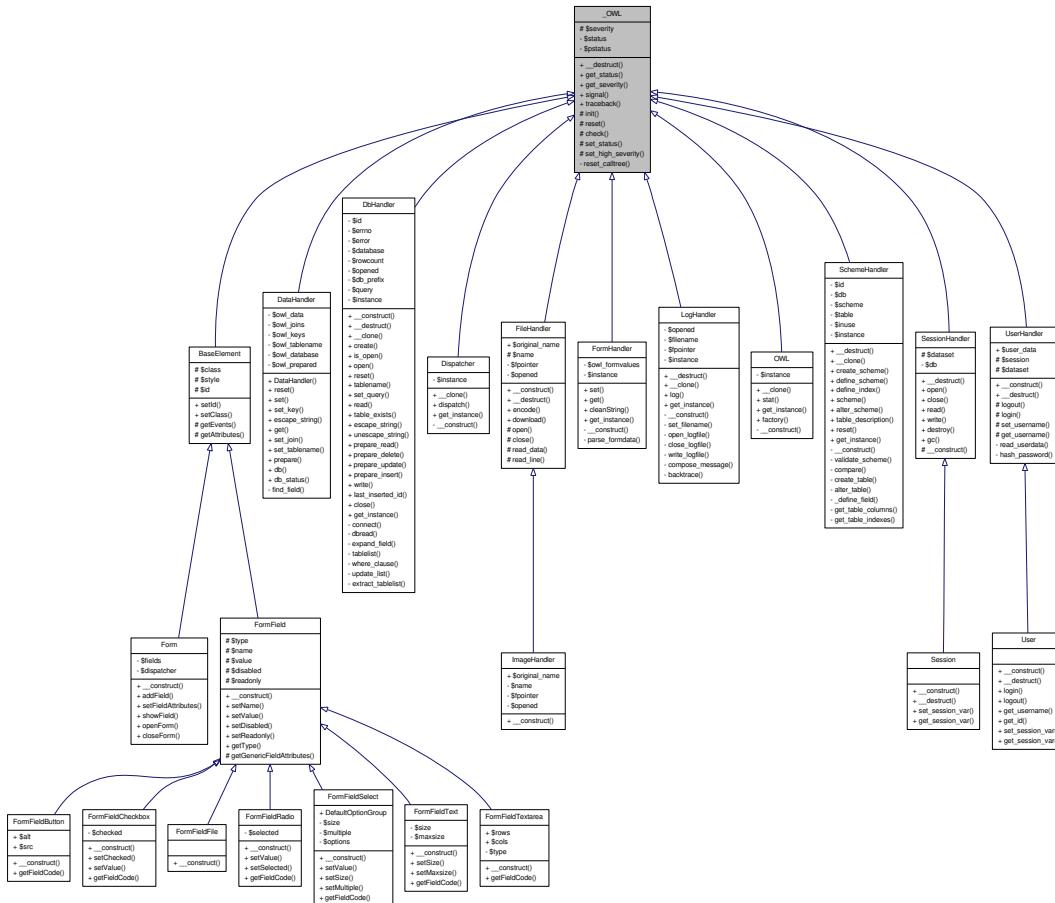
- file [owl.debug.functions.php](#)
- file [owl.helper.functions.php](#)
- file [owl.messages.en-uk.php](#)
- file [owl.messages.php](#)
- file [owl.nodebug.functions.php](#)
- file [owl.severitycodes.php](#)
- file [OWLloader.php](#)
- file [OWLRundown.php](#)

Chapter 6

Class Documentation

6.1 OWL Class Reference

Inheritance diagram for _OWL:



Public Member Functions

- `__destruct ()`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Protected Member Functions

- `init ()`
- `reset ()`
- `check (&$object, $level)`
- `set_status ($status, $params=array())`
- `set_high_severity (&$object=null)`

Protected Attributes

- `$severity`

Private Member Functions

- `reset_calltree ($depth=0)`

Private Attributes

- `$status`
- `$pstatus`

6.1.1 Detailed Description

This is the main class for all `OWL` objects. It contains some methods that have to be available to all objects. Some of them can be reimplemented.

Author

Oscar van Eijk, Oveas Functionality Provider

Version

May 15, 2007 -- O van Eijk -- initial version

6.1.2 Constructor & Destructor Documentation

6.1.2.1 `_OWL::__destruct ()`

Default class destructor; has to exist but can (should?) be reimplemented

Reimplemented in `Session`, `User`, `DbHandler`, `FileHandler`, `LogHandler`, `SchemeHandler`, `SessionHandler`, and `UserHandler`.

6.1.3 Member Function Documentation

6.1.3.1 `_OWL::check (&$ object, $ level)` [protected]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← `$object` Pointer to an object to check against
- ← `$level` The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `reset()`, `set_high_severity()`, and `traceback()`.

Referenced by `SessionHandler::write()`.

6.1.3.2 `_OWL::get_severity ($ status = null)`

Get the current object severity level.

Parameters

- ← `$status` An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References `$status`.

Referenced by `LogHandler::compose_message()`, `LogHandler::log()`, and `DataHandler::set_key()`.

6.1.3.3 `_OWL::get_status ()` [final]

Get the current object status.

Returns

Object's status code

Referenced by `SchemeHandler::compare()`.

6.1.3.4 `_OWL::init ()` [protected]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References `OWL::factory()`.

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), LogHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), OWL::__construct(), Dispatcher::__construct(), and DataHandler::DataHandler().

6.1.3.5 __OWL::reset () [protected]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References `reset_calltree()`.

Referenced by `check()`, `SessionHandler::read()`, `DataHandler::reset()`, and `set_status()`.

6.1.3.6 __OWL::reset_calltree (\$ depth = 0) [final, private]

Reset the status in the complete calltree

Parameters

$\leftarrow \$depth$ Keep track of the depth in recursive calls. Should be empty in the first call.

Referenced by `reset()`.

6.1.3.7 __OWL::set_high_severity (&\$ object = null) [protected]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.1.3.8 __OWL::set_status (\$ status, \$ params = array ()) [final, protected]

Set the current object status to the specified value.

Parameters

$\leftarrow \$status$ OWL status code

$\leftarrow \$params$

References `$GLOBALS`, `$status`, `ConfigHandler::get()`, `Register::get_code()`, `reset()`, and `signal()`.

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), Form::addField(), SchemeHandler::alter_scheme(), FileHandler::close(), DbHandler::connect(), DbHandler::create(), SchemeHandler::create_scheme(), DataHandler::DataHandler(), SchemeHandler::define_index(), SchemeHandler::define_scheme(), Dispatcher::dispatch(), FormHandler::get(), DataHandler::get(), SchemeHandler::get_table_columns(), SchemeHandler::get_table_indexes(), UserHandler::login(), FileHandler::open(), DbHandler::open(), LogHandler::open_logfile(), DataHandler::prepare(), DbHandler::prepare_delete(), DbHandler::prepare_insert(), DbHandler::prepare_read(), DbHandler::prepare_update(), DbHandler::read(), FileHandler::read_line(), UserHandler::read_userdata(), DataHandler::reset(), SchemeHandler::scheme(), FormHandler::set(), DataHandler::set(),

DataHandler::set_join(), DataHandler::set_key(), Form::setFieldAttributes(), FormFieldText::setMaxsize(), FormFieldRadio::setSelected(), FormFieldText::setSize(), FormFieldSelect::setSize(), FormFieldSelect::setValue(), FormFieldRadio::setValue(), Form::showField(), SchemeHandler::table_description(), SchemeHandler::validate_scheme(), and DbHandler::write().

6.1.3.9 `_OWL::signal ($level = OWL_INFO, &$text = false)`

Display the message for the current object status

Parameters

← **\$level** An optional severity level; message will only be displayed when it is at least of this level.

\$text If this parameter is given, the message text is returned in this string instead of echo'd.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by set_status(), and traceback().

6.1.3.10 `_OWL::traceback (&$text = false, $depth = 0)`

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References signal().

Referenced by check(), UserHandler::login(), and SessionHandler::read().

6.1.4 Member Data Documentation

6.1.4.1 `_OWL::$pstatus [private]`

Pointer to the object which holds the last nonsuccessfull (\geq OWL_WARNING) status

6.1.4.2 `_OWL::$severity [protected]`

Severity level of the current object status

6.1.4.3 `_OWL::$status` [private]

Current object status

Referenced by `get_severity()`, and `set_status()`.

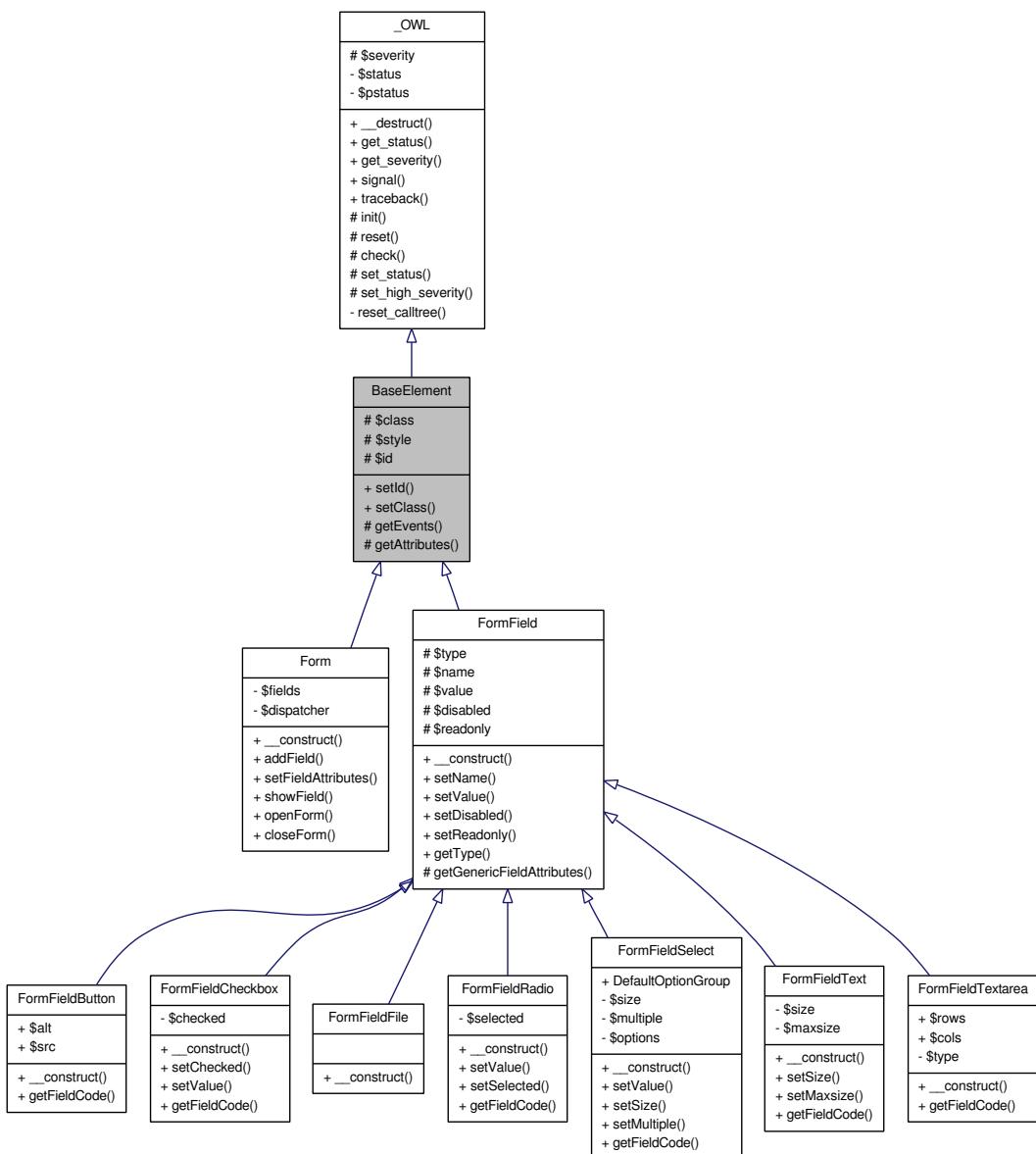
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/[class._owl.php](#)

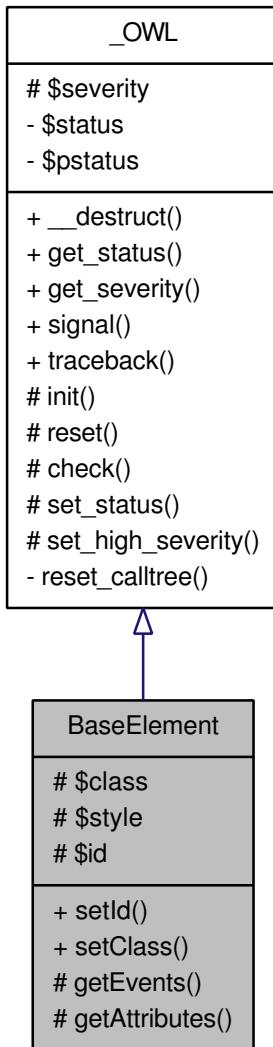
6.2 BaseElement Class Reference

DOM Element base class.

Inheritance diagram for BaseElement:



Collaboration diagram for BaseElement:



Public Member Functions

- `setId ($_value)`
- `setClass ($_value)`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Protected Member Functions

- `getEvents ()`
- `getAttributes ()`
- `init ()`

- `reset ()`
- `check (&$object, $level)`
- `set_status ($status, $params=array())`
- `set_high_severity (&$object=null)`

Protected Attributes

- `$class = "`
- `$style = "`
- `$id = "`
- `$severity`

6.2.1 Detailed Description

DOM Element base class. Abstract base class for all DOM elements

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Aug 29, 2008 -- O van Eijk -- initial version

6.2.2 Member Function Documentation

6.2.2.1 `_OWL::check (&$ object, $ level)` [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← `$object` Pointer to an object to check against
- ← `$level` The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.2.2.2 `_OWL::get_severity ($ status = null)` [inherited]

Get the current object severity level.

Parameters

- ← `$status` An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References `_OWL::$status`.

Referenced by `LogHandler::compose_message()`, `LogHandler::log()`, and `DataHandler::set_key()`.

6.2.2.3 `_OWL::get_status ()` [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by `SchemeHandler::compare()`.

6.2.2.4 `BaseElement::getAttributes ()` [protected]

Return the general element attributes that are set

Returns

string attributes in HTML format (' fld="value"...')

Referenced by `Form::openForm()`.

6.2.2.5 `BaseElement::getEvents ()` [protected]

Referenced by `FormField::getGenericFieldAttributes()`.

6.2.2.6 `_OWL::init ()` [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References `OWL::factory()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `LogHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `OWL::__construct()`, `Dispatcher::__construct()`, and `DataHandler::DataHandler()`.

6.2.2.7 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References `_OWL::reset_calltree()`.

Referenced by `_OWL::check()`, `SessionHandler::read()`, `DataHandler::reset()`, and `_OWL::set_status()`.

6.2.2.8 `_OWL::set_high_severity (&$ object = null)` [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.2.2.9 `_OWL::set_status ($ status, $ params = array ())` [final, protected, inherited]

Set the current object status to the specified value.

Parameters

$\leftarrow \$status$ OWL status code
 $\leftarrow \$params$

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler::DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`, `Dispatcher::dispatch()`, `FormHandler::get()`, `DataHandler::get()`, `SchemeHandler::get_table_columns()`, `SchemeHandler::get_table_indexes()`, `UserHandler::login()`, `FileHandler::open()`, `DbHandler::open()`, `LogHandler::open_logfile()`, `DataHandler::prepare()`, `DbHandler::prepare_delete()`, `DbHandler::prepare_insert()`, `DbHandler::prepare_read()`, `DbHandler::prepare_update()`, `DbHandler::read()`, `FileHandler::read_line()`, `UserHandler::read_userdata()`, `DataHandler::reset()`, `SchemeHandler::scheme()`, `FormHandler::set()`, `DataHandler::set()`, `DataHandler::set_join()`, `DataHandler::set_key()`, `Form::setFieldAttributes()`, `FormFieldText::setMaxsize()`, `FormFieldRadio::setSelected()`, `FormFieldText::setSize()`, `FormFieldSelect::setSize()`, `FormFieldSelect::setValue()`, `FormFieldRadio::setValue()`, `Form::showField()`, `SchemeHandler::table_description()`, `SchemeHandler::validate_scheme()`, and `DbHandler::write()`.

6.2.2.10 `BaseElement::setClass ($ _value)`

Set the element Class

Parameters

$\leftarrow \$_value$ Class name

6.2.2.11 `BaseElement::setId ($ _value)`

Set the element ID

Parameters

$\leftarrow \$_value$ Identification

6.2.2.12 `_OWL::signal ($level = OWL_INFO, &$text = false)` [inherited]

Display the message for the current object status

Parameters

$\leftarrow \$level$ An optional severity level; message will only be displayed when it is at least of this level.

$\$text$ If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by _OWL::set_status(), and _OWL::traceback().

6.2.2.13 `_OWL::traceback (&$text = false, $depth = 0)` [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

$\$text$ Optional variable in which the message text can be stored. If not given, the text will be written to standard output

$\leftarrow \$depth$ This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References _OWL::signal().

Referenced by _OWL::check(), UserHandler::login(), and SessionHandler::read().

6.2.3 Member Data Documentation

6.2.3.1 `BaseElement::$class = ''` [protected]

Class specification

6.2.3.2 `BaseElement::$id = ''` [protected]

Element ID

6.2.3.3 `_OWL::$severity` [protected, inherited]

Severity level of the current object status

6.2.3.4 BaseElement::\$style = "" [protected]

Element style

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

- /home/oscar/projects/owl-php/src/kernel/ui/[class.baseelement.php](#)

6.3 ConfigHandler Class Reference

Configuration handler.

Static Public Member Functions

- static `read_config ($file= "")`
- static `get ($item, $default=null)`
- static `set ($item, $value)`

Static Private Member Functions

- static `convert ($val)`

6.3.1 Detailed Description

Configuration handler. This abstract class reads configuration from file or database, and fills and reads the global datastructure with config items

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Aug 20, 2008 -- O van Eijk -- initial version

6.3.2 Member Function Documentation

6.3.2.1 static ConfigHandler::convert (\$ val) [static, private]

Convert values in character string format to a known value

Parameters

$\leftarrow \$val$ The value as read from the config file

Returns

Value in the desired format (or as is if nothing set)

References Register::get_severity_level().

Referenced by read_config().

6.3.2.2 static ConfigHandler::get (\$ item, \$ default = null) [static]

Return a configuration value. Note! In order to use hidden values properly, this is the ONLY way configuration values should be retrieved!

Parameters

$\leftarrow \$item$ The configuration item in the same format as it appears in the configuration file
(e.g. 'group subject item')

← **\$default** The default value to return if the config item was not set. This defaults to 'null'; if it is anything other than null, the CONFIG_NOVALUE status will not be set

Returns

Corresponding value of null when nothing was found

References \$GLOBALS, and OWL::stat().

Referenced by SessionHandler::__construct(), LogHandler::__construct(), OWLException::__construct(), DbHandler::__construct(), User::__construct(), OWLException::check_hide(), LogHandler::compose_message(), DbHandler::connect(), FormHandler::get(), DbHandler::get_instance(), UserHandler::hash_password(), LogHandler::log(), OWLExceptionHandler::log_exception(), UserHandler::login(), User::logout(), Register::register_messages(), FormHandler::set(), LogHandler::set_filename(), _OWL::set_status(), _OWL::signal(), and OWLException::traceback().

6.3.2.3 static ConfigHandler::read_config (\$file = '') [static]

Parse the given configuration file

Parameters

← **\$file** Full path to the configuration file

References \$GLOBALS, and convert().

6.3.2.4 static ConfigHandler::set (\$item, \$value) [static]

Set a configuration item. Existing values will be overwritten.

Parameters

← **\$item** The configuration item in the same format as it appears in the configuration file
(e.g. 'group subject item')

← **\$value** The new value of the item

References \$GLOBALS.

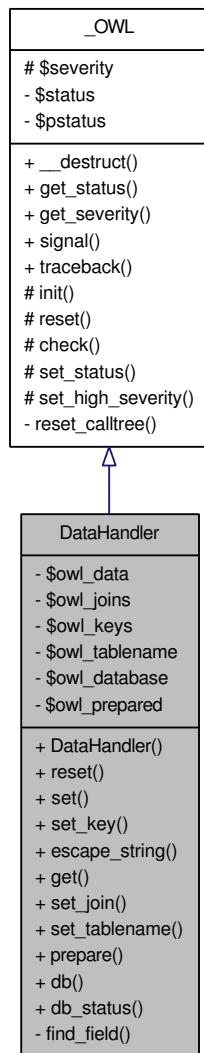
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/so/[class.confighandler.php](#)

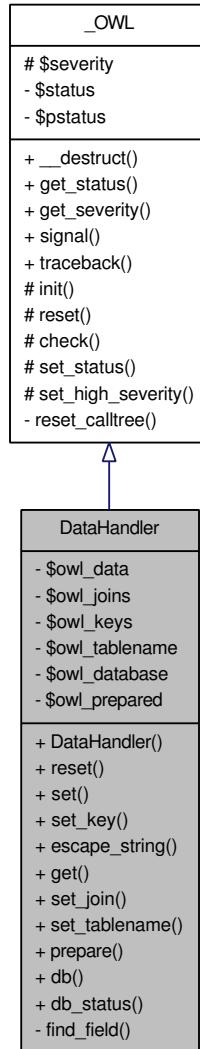
6.4 DataHandler Class Reference

The [OWL](#) Data object.

Inheritance diagram for DataHandler:



Collaboration diagram for DataHandler:



Public Member Functions

- `DataHandler ($tablename= '')`
- `reset ($level=DATA_RESET_PREPARE)`
- `set ($variable, $value)`
- `set_key ($variable)`
- `escape_string ($string)`
- `get ($variable)`
- `set_join ($lvalue, $rvalue, $linktype= '=')`
- `set_tablename ($tblname)`
- `prepare ($type=DATA_READ)`
- `db (&$data=false, $line=0, $file= '[unknown]')`
- `db_status ()`
- `get_status ()`
- `get_severity ($status=null)`

- `signal` (\$level=OWL_INFO, &\$text=false)
- `traceback` (&\$text=false, \$depth=0)

Protected Member Functions

- `init` ()
- `reset` ()
- `check` (&\$object, \$level)
- `set_status` (\$status, \$params=array())
- `set_high_severity` (&\$object=null)

Protected Attributes

- `$severity`

Private Member Functions

- `find_field` (\$fld, &\$expanded)

Private Attributes

- `$owl_data`
- `$owl_joins`
- `$owl_keys`
- `$owl_tablename`
- `$owl_database`
- `$owl_prepared`

6.4.1 Detailed Description

The OWL Data object. This class contains DB datasets

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Aug 4, 2008 -- O van Eijk -- initial version

6.4.2 Member Function Documentation

6.4.2.1 `_OWL::check (&$ object, $ level)` [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

$\leftarrow \$object$ Pointer to an object to check against

← **\$level** The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.4.2.2 DataHandler::DataHandler (`$ tablename = ''`)

Class constructor. We don't use a standard constructor (`__construct()`) here, since singleton classes (using private constructors) might derive from this class. In stead we use the PHP4 compatible constructor.

Parameters

← **\$tablename** Default table name for this dataset

References `OWL::factory()`, `_OWL::init()`, and `_OWL::set_status()`.

6.4.2.3 DataHandler::db (`&$ data = false, $ line = 0, $ file = '[unknown]'`)

Forward a call to the DBHandler object (which is private in this [DataHandler](#))

Parameters

\$data The result of DBHandler function, or false when no query was prepared yet

← **\$line** Line number of this call

← **\$file** File that made the call to this method

Returns

Severity level

References `_OWL::set_high_severity()`.

6.4.2.4 DataHandler::db_status ()

Return the last status of the database

Returns

Current status of the database object

6.4.2.5 DataHandler::escape_string (`$ string`)

Interface to the `DbHandler::escape_string()`

Parameters

← **\$string** String to escape

Returns

Return value of [DbHandler::escape_string\(\)](#)

6.4.2.6 DataHandler::find_field (\$ fld, &\$ expanded) [private]

Try to expand a field to a fully qualified 'table#field' name

Parameters

$\leftarrow \$fld$ The fieldname that has to be expanded

$\$expanded$ An array with all matching fully qualified fieldnames.

Returns

The number of matches

Referenced by [get\(\)](#).

6.4.2.7 DataHandler::get (\$ variable)

Retrieve a value from the data array. The variable name can be a fully qualified table/fieldname (format "table#field"), or only a field name, in which case it has to be unique. If the fieldname cannot be found directly, the array is scanned to find a matching field. If more matches are found, the object status is set to DATA_AMBFIELD.

Parameters

$\leftarrow \$variable$ The name of the variable that should be retrieved

Returns

The value, or NULL when the value was not found or ambiguous.

References [find_field\(\)](#), and [_OWL::set_status\(\)](#).

6.4.2.8 _OWL::get_severity (\$ status = null) [inherited]

Get the current object severity level.

Parameters

$\leftarrow \$status$ An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References [_OWL::\\$status](#).

Referenced by [LogHandler::compose_message\(\)](#), [LogHandler::log\(\)](#), and [set_key\(\)](#).

6.4.2.9 `_OWL::get_status ()` [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by SchemeHandler::compare().

6.4.2.10 `_OWL::init ()` [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References OWL::factory().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), LogHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), OWL::__construct(), Dispatcher::__construct(), and DataHandler().

6.4.2.11 `DataHandler::prepare ($ type = DATA_READ)`

Prepare a database query

Parameters

← \$type Specify which type of query should be prepared:

- DATA_READ (default); Read data from the database
- DATA_WRITE; Write new data to the database
- DATA_UPDATE; Update data in the database

Returns

Severity level

References \$_t, \$_table, _OWL::set_high_severity(), and _OWL::set_status().

6.4.2.12 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References _OWL::reset_calltree().

Referenced by _OWL::check(), SessionHandler::read(), reset(), and _OWL::set_status().

6.4.2.13 `DataHandler::reset ($ level = DATA_RESET_PREPARE)`

Reset the object

Parameters

← **\$level** The reset level, can be any of the following:

- DATA_RESET_STATUS; Reset object status only
- DATA_RESET_PREPARE (default); Reset status and a prepared query
- DATA_RESET_META; Reset status and query, remove all locks and joins
- DATA_RESET_FULL; Reset status and query, remove all locks, joins and data

References `_OWL::reset()`, and `_OWL::set_status()`.

6.4.2.14 DataHandler::set (\$ variable, \$ value)

Define or override a variable in the data array

Parameters

← **\$variable** The name of the variable that should be set

← **\$value** Value to set the variable to. If this is an array, the second field in the array has to be the tablename where the fieldname is found. The value itself van NEVER be an array!

References `_OWL::set_status()`.

6.4.2.15 _OWL::set_high_severity (&\$ object = null) [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `db()`, `prepare()`, and `SessionHandler::read()`.

6.4.2.16 DataHandler::set_join (\$ lvalue, \$ rvalue, \$ linktype = '=')

Define a link between 2 fields that will be recognized when the database query is built.

Parameters

← **\$lvalue** Left value as array(table, field)

← **\$rvalue** Right value as array(table, field)

← **\$linktype** How are the fields linked. Can be any binary operator as recognized by SQL.

Returns

Severity level

References `_OWL::set_status()`.

6.4.2.17 DataHandler::set_key (\$ variable)

Lock variables for update by adding them to an array. Fields in this array will not be overwritten on updates, but used in WHERE clauses.

Parameters

$\leftarrow \$variable$ Variable name to lock, optionally as an array (table, field)

Returns

Severity level

References `_OWL::get_severity()`, and `_OWL::set_status()`.

6.4.2.18 `_OWL::set_status ($status, $params = array ())` [final, protected, inherited]

Set the current object status to the specified value.

Parameters

$\leftarrow \$status$ OWL status code

$\leftarrow \$params$

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`, `Dispatcher::dispatch()`, `FormHandler::get()`, `get()`, `SchemeHandler::get_table_columns()`, `SchemeHandler::get_table_indexes()`, `UserHandler::login()`, `FileHandler::open()`, `DbHandler::open()`, `LogHandler::open_logfile()`, `prepare()`, `DbHandler::prepare_delete()`, `DbHandler::prepare_insert()`, `DbHandler::prepare_read()`, `DbHandler::prepare_update()`, `DbHandler::read()`, `FileHandler::read_line()`, `UserHandler::read_userdata()`, `reset()`, `SchemeHandler::scheme()`, `FormHandler::set()`, `set()`, `set_join()`, `set_key()`, `Form::setFieldAttributes()`, `FormFieldText::setMaxsize()`, `FormFieldRadio::setSelected()`, `FormFieldText::setSize()`, `FormFieldSelect::setSize()`, `FormFieldSelect::setValue()`, `FormFieldRadio::setValue()`, `Form::showField()`, `SchemeHandler::table_description()`, `SchemeHandler::validate_scheme()`, and `DbHandler::write()`.

6.4.2.19 `DataHandler::set tablename ($tblname)`

Set or overwrite the default table name

Parameters

$\leftarrow \$tblname$ Default table name

6.4.2.20 `_OWL::signal ($level = OWL_INFO, &$text = false)` [inherited]

Display the message for the current object status

Parameters

$\leftarrow \$level$ An optional severity level; message will only be displayed when it is at least of this level.

\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by _OWL::set_status(), and _OWL::traceback().

6.4.2.21 _OWL::traceback (&\$ text = false, \$ depth = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References _OWL::signal().

Referenced by _OWL::check(), UserHandler::login(), and SessionHandler::read().

6.4.3 Member Data Documentation

6.4.3.1 DataHandler::\$owl_data [private]

Indexed array holding all data values.

6.4.3.2 DataHandler::\$owl_database [private]

An optional link to a database object. This has to be specified if the data needs to be written to or read from a database.

6.4.3.3 DataHandler::\$owl_joins [private]

2D Array holding all relationships between the data.

6.4.3.4 DataHandler::\$owl_keys [private]

Array with variable names that are used in WHERE clauses on updates

6.4.3.5 DataHandler::\$owl_prepared [private]

Boolean that indicates if a query has been prepared

6.4.3.6 DataHandler::\$owl_tablename [private]

All variable names are expected to be fields in a database as well. If a table name is not given, the default table name will be used. This is useful for datasets that come from only one database table. For datasets that are not read from or written to a database, the tablename can be null.

6.4.3.7 __OWL::\$severity [protected, inherited]

Severity level of the current object status

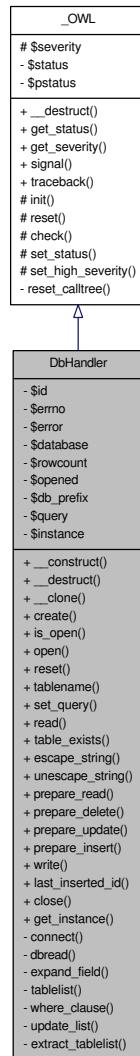
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/so/[class.datahandler.php](#)

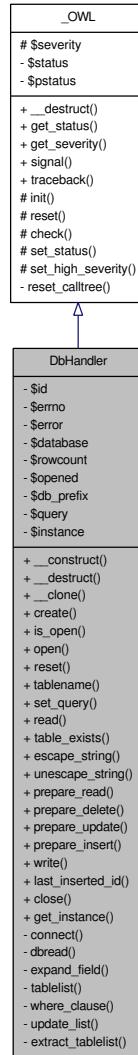
6.5 DbHandler Class Reference

Database handler.

Inheritance diagram for DbHandler:



Collaboration diagram for DbHandler:



Public Member Functions

- `__construct ($srv= 'localhost', $db= '', $usr= '', $pwd= '', $dbtype= 'MySQL')`
- `__destruct ()`
- `__clone ()`
- `create ()`
- `is_open ()`
- `open ()`
- `reset ()`
- `tablename ($tablename)`
- `set_query ($qry)`
- `read ($flag=DBHANDLE_DATA, &$data, $quick_query= '', $line=0, $file= '[unknown]')`
- `table_exists ($tablename)`
- `escape_string ($string)`
- `unescape_string ($string)`

- `prepare_read` (\$values=array(), \$tables=array(), \$searches=array(), \$joins=array())
- `prepare_delete` (\$searches=array())
- `prepare_update` (\$values=array(), \$searches=array(), \$joins=array())
- `prepare_insert` (\$values=array())
- `write` (\$rows=false, \$line=0, \$file='[unknown]')
- `last_inserted_id` ()
- `close` ()
- `get_status` ()
- `get_severity` (\$status=null)
- `signal` (\$level=OWL_INFO, &\$text=false)
- `traceback` (&\$text=false, \$depth=0)

Static Public Member Functions

- static `get_instance` ()

Protected Member Functions

- `init` ()
- `check` (&\$object, \$level)
- `set_status` (\$status, \$params=array())
- `set_high_severity` (&\$object=null)

Protected Attributes

- `$severity`

Private Member Functions

- `connect` ()
- `dbread` (\$qry, &\$rows, &\$fields)
- `expand_field` (&\$field, \$quotes=false)
- `tablelist` (\$tables)
- `where_clause` (\$searches, \$joins)
- `update_list` (\$updates)
- `extract_tablelist` (\$fields)

Private Attributes

- `$id`
- `$errno`
- `$error`
- `$database`
- `$rowcount`
- `$opened`
- `$db_prefix`
- `$query`

Static Private Attributes

- static `$instance`

6.5.1 Detailed Description

Database handler. Handler for all database I/O. This singleton class uses an (abstract) class for the actual storage.

Author

Oscar van Eijk, Oveas Functionality Provider

Version

May 15, 2007 -- O van Eijk -- initial version for Terra-Terra
Jul 29, 2008 -- O van Eijk -- Modified version for [OWL](#)

6.5.2 Constructor & Destructor Documentation

6.5.2.1 `DbHandler::__construct ($srv = 'localhost', $db = '', $usr = '', $pwd = '', $dbtype = 'MySQL')`

Class constructor; opens the database connection.

Parameters

← `$srv` Database server
← `$db` Database name
← `$usr` Username to connect with
← `$pwd` Password to use for connection
← `$dbtype` Database type (reserved for future use, currently only MySQL is implemented)

References `ConfigHandler::get()`, `_OWL::init()`, and `_OWL::set_status()`.

6.5.2.2 `DbHandler::__destruct ()`

Class destructor; closes the database connection

Reimplemented from [_OWL](#).

References `close()`.

6.5.3 Member Function Documentation

6.5.3.1 `DbHandler::__clone ()`

Implementation of the [`__clone\(\)`](#) function to prevent cloning of this singleton; it triggers a fatal (user)error

6.5.3.2 `_OWL::check (&$ object, $ level)` [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← `$object` Pointer to an object to check against
- ← `$level` The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.5.3.3 `DbHandler::close ()`

Close the database and disconnect from the server. This function is called on program shutdown. Although the database will be closed already by PHP, this function might be called at any time manually; it also updates the 'opened' variable.

Referenced by `__destruct()`, and `create()`.

6.5.3.4 `DbHandler::connect ()` [private]

Connect to the database server

Returns

True on success, otherwise False

References `ConfigHandler::get()`, and `_OWL::set_status()`.

Referenced by `create()`, and `open()`.

6.5.3.5 `DbHandler::create ()`

Create a new database

Returns

Severity level

References `close()`, `connect()`, and `_OWL::set_status()`.

6.5.3.6 `DbHandler::dbread ($ qry, &$ rows, &$ fields)` [private]

Read from the database.

Parameters

← **\$qry** Database query string.
← **\$rows** Number of rows matched
← **\$fields** Number of fields per row

Returns

A 2D array with all data, or false on failures

Referenced by `read()`, and `table_exists()`.

6.5.3.7 DbHandler::escape_string (\$ string)

Call the current DBtype's escape function for character strings.

Parameters

\$string The string that should be escaped

Returns

The escaped string

6.5.3.8 DbHandler::expand_field (&\$ field, \$ quotes = false) [private]

Change a fieldname in the format 'table#field' to the format '[prefix]table.field'

Parameters

← **\$field** Fieldname to expand
← **\$quotes** If true, add backquotes

References `$_t`, and `tablename()`.

Referenced by `prepare_insert()`, `prepare_read()`, `update_list()`, and `where_clause()`.

6.5.3.9 DbHandler::extract_tablelist (\$ fields) [private]

Create an array with unique tablenames as extracted from an array of fields in the format (table::field => value, ...)

Parameters

← **\$fields** An array with fields

Returns

Array with tablenames

References `$_t`, and `$_table`.

Referenced by `prepare_delete()`, `prepare_insert()`, and `prepare_update()`.

6.5.3.10 static DbHandler::get_instance () [static]

Return a reference to my implementation. If necessary, create that implementation first.

Returns

Object instance ID

References \$instance, and ConfigHandler::get().

6.5.3.11 _OWL::get_severity (\$status = null) [inherited]

Get the current object severity level.

Parameters

← *\$status* An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References _OWL::\$status.

Referenced by LogHandler::compose_message(), LogHandler::log(), and DataHandler::set_key().

6.5.3.12 _OWL::get_status () [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by SchemeHandler::compare().

6.5.3.13 _OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References OWL::factory().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), LogHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), __construct(), OWL::__construct(), Dispatcher::__construct(), and DataHandler::DataHandler().

6.5.3.14 DbHandler::is_open ()

Let other objects check if th database connection is opened

Returns

boolean, True when opened

6.5.3.15 DbHandler::last_inserted_id ()

Return the last ID after a newly inserted record holding an AUTO_INCREMENT field

Returns

The number that was last inserted

6.5.3.16 DbHandler::open ()

Opens the database connection.

Returns

Severity level

References connect(), and _OWL::set_status().

Referenced by read().

6.5.3.17 DbHandler::prepare_delete (\$searches = array())

Prepare a delete query. Data is taken from the arrays that are passed to this function. All fieldnames are in the format 'table#field', where the table is not yet prefixed.

Parameters

← *\$searches* Given values that have to match

Returns

Severity level

References extract_tablelist(), _OWL::set_status(), tablelist(), and where_clause().

6.5.3.18 DbHandler::prepare_insert (\$values = array())

Prepare an insert query. Data is taken from the arrays that are passed to this function. All fieldnames are in the format 'table#field', where the table is not yet prefixed.

Parameters

← *\$values* Given database values

Returns

Severity level

References expand_field(), extract_tablelist(), _OWL::set_status(), and tablename().

6.5.3.19 DbHandler::prepare_read (\$ values = array(), \$ tables = array(), \$ searches = array(), \$ joins = array())

Prepare a read query. Data is taken from the arrays that are passed to this function. All fieldnames are in the format 'table#field', where the table is not yet prefixed.

Parameters

- ← **\$values** Values that will be read
- ← **\$tables** Tables from which will be read
- ← **\$searches** Given values that have to match
- ← **\$joins** Joins on the given tables

Returns

Severity level

References expand_field(), _OWL::set_status(), tablelist(), and where_clause().

6.5.3.20 DbHandler::prepare_update (\$ values = array(), \$ searches = array(), \$ joins = array())

Prepare an update query. Data is taken from the arrays that are passed to this function. All fieldnames are in the format 'table#field', where the table is not yet prefixed.

Parameters

- ← **\$values** Given database values
- ← **\$searches** List of fieldnames that will be used in the where clause. All fields not in this array will be updated!
- ← **\$joins** Joins on the given tables

Returns

Severity level

References extract_tablelist(), _OWL::set_status(), tablelist(), update_list(), and where_clause().

6.5.3.21 DbHandler::read (\$ flag = DBHANDLE_DATA, &\$ data, \$ quick_query = '', \$ line = 0, \$ file = '[unknown]')

Read from the database. The return value depends on the flag. By default, the selected rows(s) are returned in a 2d array.

Parameters

- ← **\$flag** Flag that identifies how data should be returned; as data (default) or the number of rows
- ← **\$data** The retrieved value in a formatt depending on the flag:
 - DBHANDLE_ROWCOUNT; Number of matching rows
 - DBHANDLE_FIELDCOUNT; Number of fields per rows

- DBHANDLE_TOTALFIELDCOUNT; Total number op fields
- DBHANDLE_DATA (default); A 2D array with all data
- DBHANDLE_SINGLEROW; The first matching row in a 1D array
- DBHANDLE_SINGLEFIELD; The first matching field

← *\$quick_query* Database query string. If empty, \$this->query is used
 ← *\$line* Line number of this call
 ← *\$file* File that made the call to this method

Returns

Severity level

References dbread(), open(), and _OWL::set_status().

6.5.3.22 DbHandler::reset ()

Reset the object

Reimplemented from [_OWL](#).

6.5.3.23 _OWL::set_high_severity (&\$ object = null) [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by _OWL::check(), DataHandler::db(), DataHandler::prepare(), and SessionHandler::read().

6.5.3.24 DbHandler::set_query (\$ qry)

Set the database query. This function should only be used if the query is too complex to be set with any of the prepare() functions.

Parameters

← *\$qry* A complete database query. All tablenames must be prefixed (with the tablename() function)!

6.5.3.25 _OWL::set_status (\$ status, \$ params = array ()) [final, protected, inherited]

Set the current object status to the specified value.

Parameters

← *\$status* [OWL](#) status code
 ← *\$params*

References \$GLOBALS, _OWL::\$status, ConfigHandler::get(), Register::get_code(), _OWL::reset(), and _OWL::signal().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), __construct(), Form::addField(), SchemeHandler::alter_scheme(), FileHandler::close(), __connect(), create(), SchemeHandler::create_scheme(), DataHandler::DataHandler(), SchemeHandler::define_index(), SchemeHandler::define_scheme(), Dispatcher::dispatch(), FormHandler::get(), DataHandler::get(), SchemeHandler::get_table_columns(), SchemeHandler::get_table_indexes(), UserHandler::login(), FileHandler::open(), open(), LogHandler::open_logfile(), DataHandler::prepare(), prepare_delete(), prepare_insert(), prepare_read(), prepare_update(), read(), FileHandler::read_line(), UserHandler::read_userdata(), DataHandler::reset(), SchemeHandler::scheme(), FormHandler::set(), DataHandler::set(), DataHandler::set_join(), DataHandler::set_key(), Form::setFieldAttributes(), FormFieldText::setMaxsize(), FormFieldRadio::setSelected(), FormFieldText::setSize(), FormFieldSelect::setSize(), FormFieldSelect::setValue(), FormFieldRadio::setValue(), Form::showField(), SchemeHandler::table_description(), SchemeHandler::validate_scheme(), and write().

6.5.3.26 _OWL::signal (\$level = OWL_INFO, &\$text = false) [inherited]

Display the message for the current object status

Parameters

← **\$level** An optional severity level; message will only be displayed when it is at least of this level.

\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by _OWL::set_status(), and _OWL::traceback().

6.5.3.27 DbHandler::table_exists (\$tablename)

Check if a table exists in the database

Parameters

← **\$tablename** Name of the table to check

Returns

True if the table exists

References dbread(), and tablename().

6.5.3.28 DbHandler::tablelist (\$tables) [private]

Create a list with tables, including prefixes, that can be interpreted by SQL

Parameters

← **\$tables** An array with tablenames

Returns

The tablelist

References `$_table`, and `tablename()`.

Referenced by `prepare_delete()`, `prepare_read()`, and `prepare_update()`.

6.5.3.29 DbHandler::tablename (\$ tablename)

Extend a tablename with the database prefix

Parameters

← **\$tablename** Table name to extend

Returns

Extended table name

Referenced by `expand_field()`, `prepare_insert()`, `table_exists()`, and `tablelist()`.

6.5.3.30 _OWL::traceback (&\$ text = false, \$ depth = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References `_OWL::signal()`.

Referenced by `_OWL::check()`, `UserHandler::login()`, and `SessionHandler::read()`.

6.5.3.31 DbHandler::unescape_string (\$ string)

Call the current DBtype's unescape function for character strings.

Parameters

\$string The string that should be unescaped

Returns

The unescaped string

6.5.3.32 DbHandler::update_list (\$updates) [private]

Create a string with 'field = value, ...' combinations in SQL format

Parameters

← *\$updates* Array with fields to update (fieldname => values)

Returns

The UPDATE statement

References expand_field().

Referenced by prepare_update().

6.5.3.33 DbHandler::where_clause (\$searches, \$joins) [private]

Create a WHERE clause that can be interpreted by SQL

Parameters

← *\$searches* Array with values (fieldname => values)

← *\$joins* Array of arrays with values (field, linktype, field)

Returns

The WHERE clause

References expand_field().

Referenced by prepare_delete(), prepare_read(), and prepare_update().

6.5.3.34 DbHandler::write (\$rows = false, \$line = 0, \$file = '[unknown]')

Database inserts and updates. The number of affected rows is stored in \$this->rowcount

Parameters

\$rows (optional) The number of affected rows

← *\$line* Line number of this call

← *\$file* File that made the call to this method

Returns

Severity level

References _OWL::set_status().

6.5.4 Member Data Documentation

6.5.4.1 DbHandler::\$database [private]

array - Database location and authorization info

6.5.4.2 DbHandler::\$db_prefix [private]

string - database prefix

6.5.4.3 DbHandler::\$errno [private]

integer - Error ID

6.5.4.4 DbHandler::\$error [private]

string - Error text

6.5.4.5 DbHandler::\$id [private]

integer - DB Handle ID

6.5.4.6 DbHandler::\$instance [static, private]

integer - self reference

Referenced by `get_instance()`.

6.5.4.7 DbHandler::\$opened [private]

boolean - True if the database is opened

6.5.4.8 DbHandler::\$query [private]

string - Query string

6.5.4.9 DbHandler::\$rowcount [private]

integer - Row counter

6.5.4.10 _OWL::\$severity [protected, inherited]

Severity level of the current object status

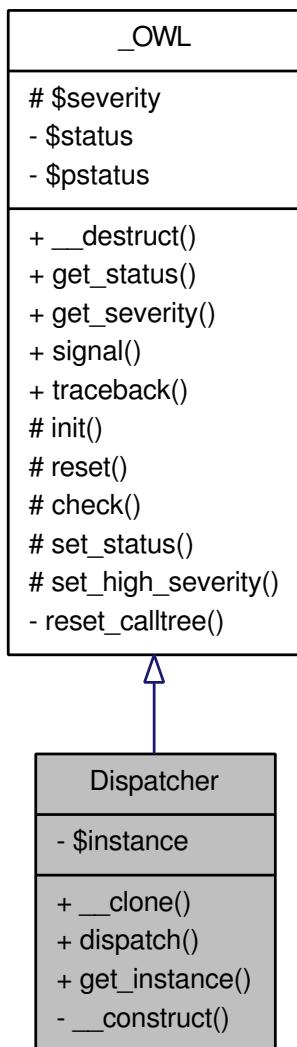
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/so/[class.dbhandler.php](#)

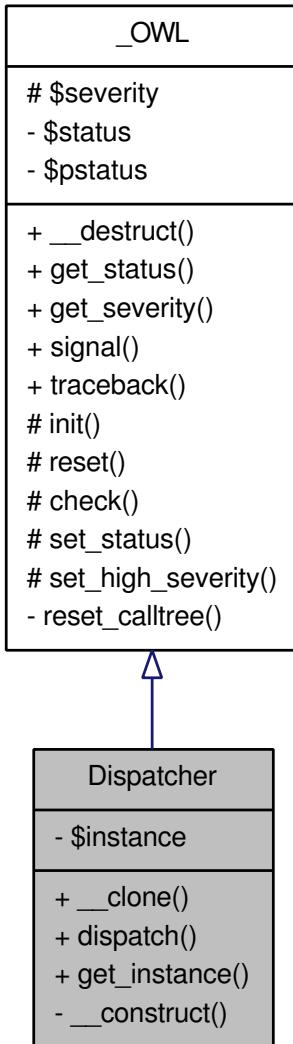
6.6 Dispatcher Class Reference

[Dispatcher](#) singleton.

Inheritance diagram for Dispatcher:



Collaboration diagram for Dispatcher:



Public Member Functions

- `__clone ()`
- `dispatch ()`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Static Public Member Functions

- static `get_instance ()`

Protected Member Functions

- `init ()`
- `reset ()`
- `check (&$object, $level)`
- `set_status ($status, $params=array())`
- `set_high_severity (&$object=null)`

Protected Attributes

- `$severity`

Private Member Functions

- `__construct ()`

Static Private Attributes

- static `$instance`

6.6.1 Detailed Description

`Dispatcher` singleton. Define the dispatcher. This class calls the proper method based in the request or form data

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Nov 23, 2010 -- O van Eijk -- initial version

6.6.2 Constructor & Destructor Documentation

6.6.2.1 Dispatcher::__construct () [private]

Constructor

References `_OWL::init()`.

6.6.3 Member Function Documentation

6.6.3.1 Dispatcher::__clone ()

Implementation of the `__clone()` function to prevent cloning of this singleton; it triggers a fatal (user)error

6.6.3.2 _OWL::check (&\$ object, \$ level) [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← *\$object* Pointer to an object to check against
- ← *\$level* The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References _OWL::reset(), _OWL::set_high_severity(), and _OWL::traceback().

Referenced by SessionHandler::write().

6.6.3.3 Dispatcher::dispatch ()

Get the dispatcher info from the formdata, load the specified classfile and call the method specified.

Returns

On errors during dispatch, the severity level, otherwise the return value of the given method

References \$_form, OWL::factory(), OWLloader::getClass(), and _OWL::set_status().

6.6.3.4 static Dispatcher::get_instance () [static]

Return a reference to my implementation. If necessary, create that implementation first.

Returns

Severity level

References \$instance.

6.6.3.5 _OWL::get_severity (\$ status = null) [inherited]

Get the current object severity level.

Parameters

- ← *\$status* An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References _OWL::\$status.

Referenced by LogHandler::compose_message(), LogHandler::log(), and DataHandler::set_key().

6.6.3.6 `_OWL::get_status ()` [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by `SchemeHandler::compare()`.

6.6.3.7 `_OWL::init ()` [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References `OWL::factory()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `LogHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `OWL::__construct()`, `__construct()`, and `DataHandler::DataHandler()`.

6.6.3.8 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in `DbHandler`, and `SchemeHandler`.

References `_OWL::reset_calltree()`.

Referenced by `_OWL::check()`, `SessionHandler::read()`, `DataHandler::reset()`, and `_OWL::set_status()`.

6.6.3.9 `_OWL::set_high_severity (&$ object = null)` [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.6.3.10 `_OWL::set_status ($ status, $ params = array ())` [final, protected, inherited]

Set the current object status to the specified value.

Parameters

$\leftarrow \$status$ `OWL` status code

$\leftarrow \$params$

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), Form::addField(), SchemeHandler::alter_scheme(), FileHandler::close(), DbHandler::connect(), DbHandler::create(), SchemeHandler::create_scheme(), DataHandler::DataHandler(), SchemeHandler::define_index(), SchemeHandler::define_scheme(), dispatch(), FormHandler::get(), DataHandler::get(), SchemeHandler::get_table_columns(), SchemeHandler::get_table_indexes(), UserHandler::login(), FileHandler::open(), DbHandler::open(), LogHandler::open_logfile(), DataHandler::prepare(), DbHandler::prepare_delete(), DbHandler::prepare_insert(), DbHandler::prepare_read(), DbHandler::prepare_update(), DbHandler::read(), FileHandler::read_line(), UserHandler::read_userdata(), DataHandler::reset(), SchemeHandler::scheme(), FormHandler::set(), DataHandler::set(), DataHandler::set_join(), DataHandler::set_key(), Form::setFieldAttributes(), FormFieldText::setMaxsize(), FormFieldRadio::setSelected(), FormFieldText::setSize(), FormFieldSelect::setSize(), FormFieldSelect::setValue(), FormFieldRadio::setValue(), Form::showField(), SchemeHandler::table_description(), SchemeHandler::validate_scheme(), and DbHandler::write().

6.6.3.11 __OWL::signal (\$level = OWL_INFO, &\$text = false) [inherited]

Display the message for the current object status

Parameters

← **\$level** An optional severity level; message will only be displayed when it is at least of this level.

\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by __OWL::set_status(), and __OWL::traceback().

6.6.3.12 __OWL::traceback (&\$text = false, \$depth = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References __OWL::signal().

Referenced by __OWL::check(), UserHandler::login(), and SessionHandler::read().

6.6.4 Member Data Documentation

6.6.4.1 Dispatcher::\$instance [static, private]

integer - self reference

Referenced by `get_instance()`.

6.6.4.2 __OWL::\$severity [protected, inherited]

Severity level of the current object status

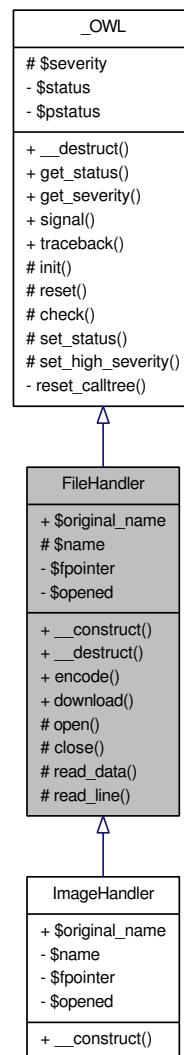
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/bo/[class.dispatcher.php](#)

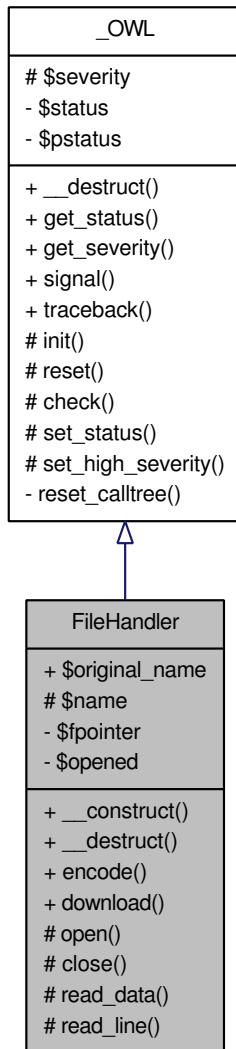
6.7 FileHandler Class Reference

File handler.

Inheritance diagram for FileHandler:



Collaboration diagram for FileHandler:



Public Member Functions

- `__construct` (`$name, $req=false`)
- `__destruct` ()
- `encode` ()
- `download` ()
- `get_status` ()
- `get_severity` (`$status=null`)
- `signal` (`$level=OWL_INFO, &$text=false`)
- `traceback` (`&$text=false, $depth=0`)

Public Attributes

- `$original_name`

Protected Member Functions

- [open](#) (\$mode= 'r')
- [close](#) ()
- [read_data](#) ()
- [read_line](#) (\$trim=FILE_NOTRIM)
- [init](#) ()
- [reset](#) ()
- [check](#) (&\$object, \$level)
- [set_status](#) (\$status, \$params=array())
- [set_high_severity](#) (&\$object=null)

Protected Attributes

- [\\$name](#)
- [\\$severity](#)

Private Attributes

- [\\$fpointer](#)
- [\\$opened](#)

6.7.1 Detailed Description

File handler. Handle all files

Author

Oscar van Eijk, Oveas Functionality Provider

Version

May 15, 2007 -- O van Eijk -- initial version for Terra-Terra (based on an old OFM module)
Jul 30, 2008 -- O van Eijk -- Modified version for OWL-PHP

6.7.2 Constructor & Destructor Documentation

6.7.2.1 FileHandler::__construct (\$ name, \$ req = false)

Class constructor; setup the file characteristics

Parameters

← [\\$name](#) Filename

← [\\$req](#) True is the file must exist at object create time

Reimplemented in [ImageHandler](#).

References [\\$name](#), [_OWL::init\(\)](#), and [_OWL::set_status\(\)](#).

6.7.2.2 FileHandler::__destruct ()

Default class destructor; has to exist but can (should?) be reimplemented

Reimplemented from [_OWL](#).

References `close()`.

6.7.3 Member Function Documentation

6.7.3.1 _OWL::check (&\$ object, \$ level) [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

← *\$object* Pointer to an object to check against

← *\$level* The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.7.3.2 FileHandler::close () [protected]

Close the file

References `_OWL::set_status()`.

Referenced by `__destruct()`, and `read_data()`.

6.7.3.3 FileHandler::download ()

6.7.3.4 FileHandler::encode ()

References `read_data()`.

6.7.3.5 _OWL::get_severity (\$ status = null) [inherited]

Get the current object severity level.

Parameters

← *\$status* An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References `_OWL::$status`.

Referenced by `LogHandler::compose_message()`, `LogHandler::log()`, and `DataHandler::set_key()`.

6.7.3.6 `_OWL::get_status () [final, inherited]`

Get the current object status.

Returns

Object's status code

Referenced by `SchemeHandler::compare()`.

6.7.3.7 `_OWL::init () [protected, inherited]`

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References `OWL::factory()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `LogHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `__construct()`, `DbHandler::__construct()`, `OWL::__construct()`, `Dispatcher::__construct()`, and `DataHandler::DataHandler()`.

6.7.3.8 `FileHandler::open ($ mode = 'r') [protected]`

Open the file

Parameters

← `$mode` Mode in which the file should be opened:

- 'r' Open for reading only; place the file pointer at the beginning of the file.
- 'r+' Open for reading and writing; place the file pointer at the beginning of the file.
- 'w' Open for writing only; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.
- 'w+' Open for reading and writing; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.
- 'a' Open for writing only; place the file pointer at the end of the file. If the file does not exist, attempt to create it.
- 'a+' Open for reading and writing; place the file pointer at the end of the file. If the file does not exist, attempt to create it.

References `_OWL::set_status()`.

Referenced by `read_data()`.

6.7.3.9 `FileHandler::read_data () [protected]`

Read the file contents and return as one dataset

References `close()`, and `open()`.

Referenced by `encode()`.

6.7.3.10 FileHandler::read_line (\$ trim = FILE_NOTRIM) [protected]

Read a single line from the file. File must be opened before

Parameters

← *\$trim* specify how the returned line should be trimmed:

- FILE_NOTRIM
- FILE_TRIM_L
- FILE_TRIM_R
- FILE_TRIM_C

References _OWL::set_status().

6.7.3.11 _OWL::reset () [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References _OWL::reset_calltree().

Referenced by _OWL::check(), SessionHandler::read(), DataHandler::reset(), and _OWL::set_status().

6.7.3.12 _OWL::set_high_severity (&\$ object = null) [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by _OWL::check(), DataHandler::db(), DataHandler::prepare(), and SessionHandler::read().

6.7.3.13 _OWL::set_status (\$ status, \$ params = array ()) [final, protected, inherited]

Set the current object status to the specified value.

Parameters

← *\$status* OWL status code

← *\$params*

References \$GLOBALS, _OWL::\$status, ConfigHandler::get(), Register::get_code(), _OWL::reset(), and _OWL::signal().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), __construct(), DbHandler::__construct(), Form::addField(), SchemeHandler::alter_scheme(), close(), DbHandler::connect(), DbHandler::create(), SchemeHandler::create_scheme(), DataHandler::DataHandler(), SchemeHandler::define_index(), SchemeHandler::define_scheme(), Dispatcher::dispatch(), FormHandler::get(), DataHandler::get(), SchemeHandler::get_table_columns(), SchemeHandler::get_table_indexes(), UserHandler::login(), open(), DbHandler::open(), LogHandler::open_logfile(), DataHandler::prepare(), DbHandler::prepare_delete(), DbHandler::prepare_insert(), DbHandler::prepare_read(), DbHandler::prepare_update(), DbHandler::read(), read_line(), UserHandler::read_userdata(), DataHandler::reset(),

SchemeHandler::scheme(), FormHandler::set(), DataHandler::set(), DataHandler::set_join(), DataHandler::set_key(), Form::setFieldAttributes(), FormFieldText::setMaxsize(), FormFieldRadio::setSelected(), FormFieldText::setSize(), FormFieldSelect::setSize(), FormFieldSelect::setValue(), FormFieldRadio::setValue(), Form::showField(), SchemeHandler::table_description(), SchemeHandler::validate_scheme(), and DbHandler::write().

6.7.3.14 `_OWL::signal ($level = OWL_INFO, &$text = false)` [inherited]

Display the message for the current object status

Parameters

← **\$level** An optional severity level; message will only be displayed when it is at least of this level.

← **\$text** If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by _OWL::set_status(), and _OWL::traceback().

6.7.3.15 `_OWL::traceback (&$text = false, $depth = 0)` [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

← **\$text** Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References _OWL::signal().

Referenced by _OWL::check(), UserHandler::login(), and SessionHandler::read().

6.7.4 Member Data Documentation

6.7.4.1 `FileHandler::$pointer` [private]

Pointer to the file when opened

Reimplemented in [ImageHandler](#).

6.7.4.2 FileHandler::\$name [protected]

Full filename as stored on the file system

Reimplemented in [ImageHandler](#).

Referenced by `__construct()`.

6.7.4.3 FileHandler::\$opened [private]

Boolean that's true when the file is opened

Reimplemented in [ImageHandler](#).

6.7.4.4 FileHandler::\$original_name

Reimplemented in [ImageHandler](#).

6.7.4.5 _OWL::\$severity [protected, inherited]

Severity level of the current object status

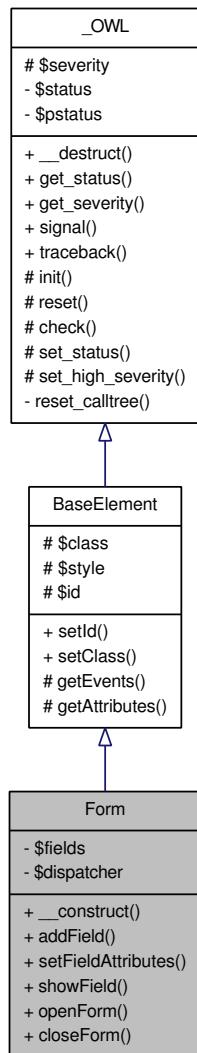
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/so/[class.filehandler.php](#)

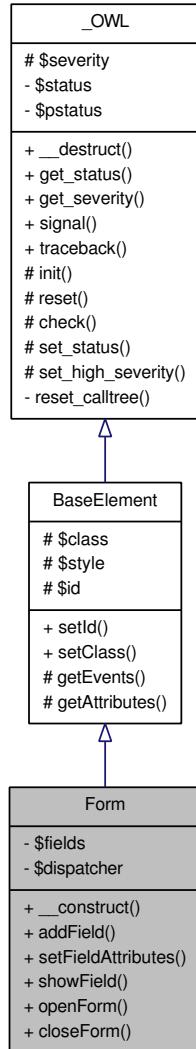
6.8 Form Class Reference

[Form](#) Element class.

Inheritance diagram for Form:



Collaboration diagram for Form:



Public Member Functions

- **__construct** (\$dispatcher)
- **addField** (\$type, \$name, \$value= "", \$attributes=array())
- **setFieldAttributes** (\$index, \$attributes)
- **showField** (\$name)
- **openForm** ()
- **closeForm** ()
- **setId** (\$value)
- **setClass** (\$value)
- **get_status** ()
- **get_severity** (\$status=null)
- **signal** (\$level=OWL_INFO, &\$text=false)
- **traceback** (&\$text=false, \$depth=0)

Protected Member Functions

- `getEvents ()`
- `getAttributes ()`
- `init ()`
- `reset ()`
- `check (&$object, $level)`
- `set_status ($status, $params=array())`
- `set_high_severity (&$object=null)`

Protected Attributes

- `$class = "`
- `$style = "`
- `$id = "`
- `$severity`

Private Attributes

- `$fields`
- `$dispatcher`

6.8.1 Detailed Description

[Form](#) Element class. Define an HTML [Form](#).

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Aug 29, 2008 -- O van Eijk -- initial version

6.8.2 Constructor & Destructor Documentation

6.8.2.1 Form::__construct (\$ dispatcher)

Class constructor;

Parameters

← `$dispatcher Dispatcher` in the format "applic#include-path#classfile#class#method", where 'include-path' can be a path relative from the applic toplevel, or a constant Classfile can be the full file name ("class.myclass.php") or just the name ("myclass") Classname can be ommitted if it is equal to the classfile starting with a capital ("Myclass")

References `$dispatcher`.

6.8.3 Member Function Documentation

6.8.3.1 Form::addField (\$ type, \$ name, \$ value = '', \$ attributes = array())

Add a formfield to the formelement

Parameters

- ← **\$type** Field type
- ← **\$name** Field name
- ← **\$value** Optional field value. For a selectlist type this must be an array, see [FormFieldSelect::setValue\(\)](#)
- ← **\$attributes** Indexed array with additional values in the format, where the key must be a supported attributed for the given type.

Returns

Severity level of the object status

References OWLloader::getClass(), _OWL::set_status(), and setFieldAttributes().

Referenced by closeForm().

6.8.3.2 _OWL::check (&\$ object, \$ level) [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← **\$object** Pointer to an object to check against
- ← **\$level** The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References _OWL::reset(), _OWL::set_high_severity(), and _OWL::traceback().

Referenced by SessionHandler::write().

6.8.3.3 Form::closeForm ()

Close the form and set a hidden field defining the dispatcher

Returns

HTML code

References addField(), and showField().

6.8.3.4 _OWL::get_severity (\$ status = null) [inherited]

Get the current object severity level.

Parameters

← **\$status** An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References _OWL::\$status.

Referenced by LogHandler::compose_message(), LogHandler::log(), and DataHandler::set_key().

6.8.3.5 _OWL::get_status () [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by SchemeHandler::compare().

6.8.3.6 BaseElement::getAttributes () [protected, inherited]

Return the general element attributes that are set

Returns

string attributes in HTML format (' fld="value"...)

Referenced by openForm().

6.8.3.7 BaseElement::getEvents () [protected, inherited]

Referenced by FormField::getGenericFieldAttributes().

6.8.3.8 _OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References OWL::factory().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), LogHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), OWL::__construct(), Dispatcher::__construct(), and DataHandler::DataHandler().

6.8.3.9 Form::openForm ()

Return the form code to open the form

Returns

HTML code

References BaseElement::getAttributes().

6.8.3.10 _OWL::reset () [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References _OWL::reset_calltree().

Referenced by _OWL::check(), SessionHandler::read(), DataHandler::reset(), and _OWL::set_status().

6.8.3.11 _OWL::set_high_severity (&\$ object = null) [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by _OWL::check(), DataHandler::db(), DataHandler::prepare(), and SessionHandler::read().

6.8.3.12 _OWL::set_status (\$ status, \$ params = array ()) [final, protected, inherited]

Set the current object status to the specified value.

Parameters

← **\$status** OWL status code
 ← **\$params**

References \$GLOBALS, _OWL::\$status, ConfigHandler::get(), Register::get_code(), _OWL::reset(), and _OWL::signal().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), addField(), SchemeHandler::alter_scheme(), FileHandler::close(), DbHandler::connect(), DbHandler::create(), SchemeHandler::create_scheme(), DataHandler::DataHandler(), SchemeHandler::define_index(), SchemeHandler::define_scheme(), Dispatcher::dispatch(), FormHandler::get(), DataHandler::get(), SchemeHandler::get_table_columns(), SchemeHandler::get_table_indexes(), UserHandler::login(), FileHandler::open(), DbHandler::open(), LogHandler::open_logfile(), DataHandler::prepare(), DbHandler::prepare_delete(), DbHandler::prepare_insert(), DbHandler::prepare_read(), DbHandler::prepare_update(), DbHandler::read(), FileHandler::read_line(), UserHandler::read_userdata(), DataHandler::reset(), SchemeHandler::scheme(), FormHandler::set(), DataHandler::set(), DataHandler::set_join(), DataHandler::set_key(), setFieldAttributes(), FormFieldText::setMaxsize(), FormFieldRadio::setSelected(), FormFieldText::setSize(), FormFieldSelect::setSize(), FormFieldSelect::setValue(), FormFieldRadio::setValue(), showField(),

SchemeHandler::table_description(), SchemeHandler::validate_scheme(), and DbHandler::write().

6.8.3.13 BaseElement::setClass (\$ _ value) [inherited]

Set the element Class

Parameters

← *\$value* Class name

6.8.3.14 Form::setFieldAttributes (\$ index, \$ attributes)

Set one or more formfield attributes

Parameters

← *\$index* Index of the fieldobject

← *\$attributes* array with object name and values in the format ('attrib' => 'value')

References _OWL::set_status().

Referenced by addField().

6.8.3.15 BaseElement::setId (\$ _ value) [inherited]

Set the element ID

Parameters

← *\$value* Identification

6.8.3.16 Form::showField (\$ name)

Get the HTML code for a given field

Parameters

← *\$name* Fieldname

Returns

HTML code defining the field

References _OWL::set_status().

Referenced by closeForm().

6.8.3.17 _OWL::signal (\$ level = OWL_INFO, &\$ text = false) [inherited]

Display the message for the current object status

Parameters

← ***\$level*** An optional severity level; message will only be displayed when it is at least of this level.

\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by _OWL::set_status(), and _OWL::traceback().

6.8.3.18 _OWL::traceback (&*\$text = false*, *\$depth = 0*) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← ***\$depth*** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References _OWL::signal().

Referenced by _OWL::check(), UserHandler::login(), and SessionHandler::read().

6.8.4 Member Data Documentation**6.8.4.1 BaseElement::\$class = ” [protected, inherited]**

Class specification

6.8.4.2 Form::\$dispatcher [private]

String holding dispatch info.

Referenced by __construct().

6.8.4.3 Form::\$fields [private]

Array holding all field objects

6.8.4.4 BaseElement::\$id = ” [protected, inherited]

Element ID

6.8.4.5 OWL::\$severity [protected, inherited]

Severity level of the current object status

6.8.4.6 BaseElement::\$style = ” [protected, inherited]

Element style

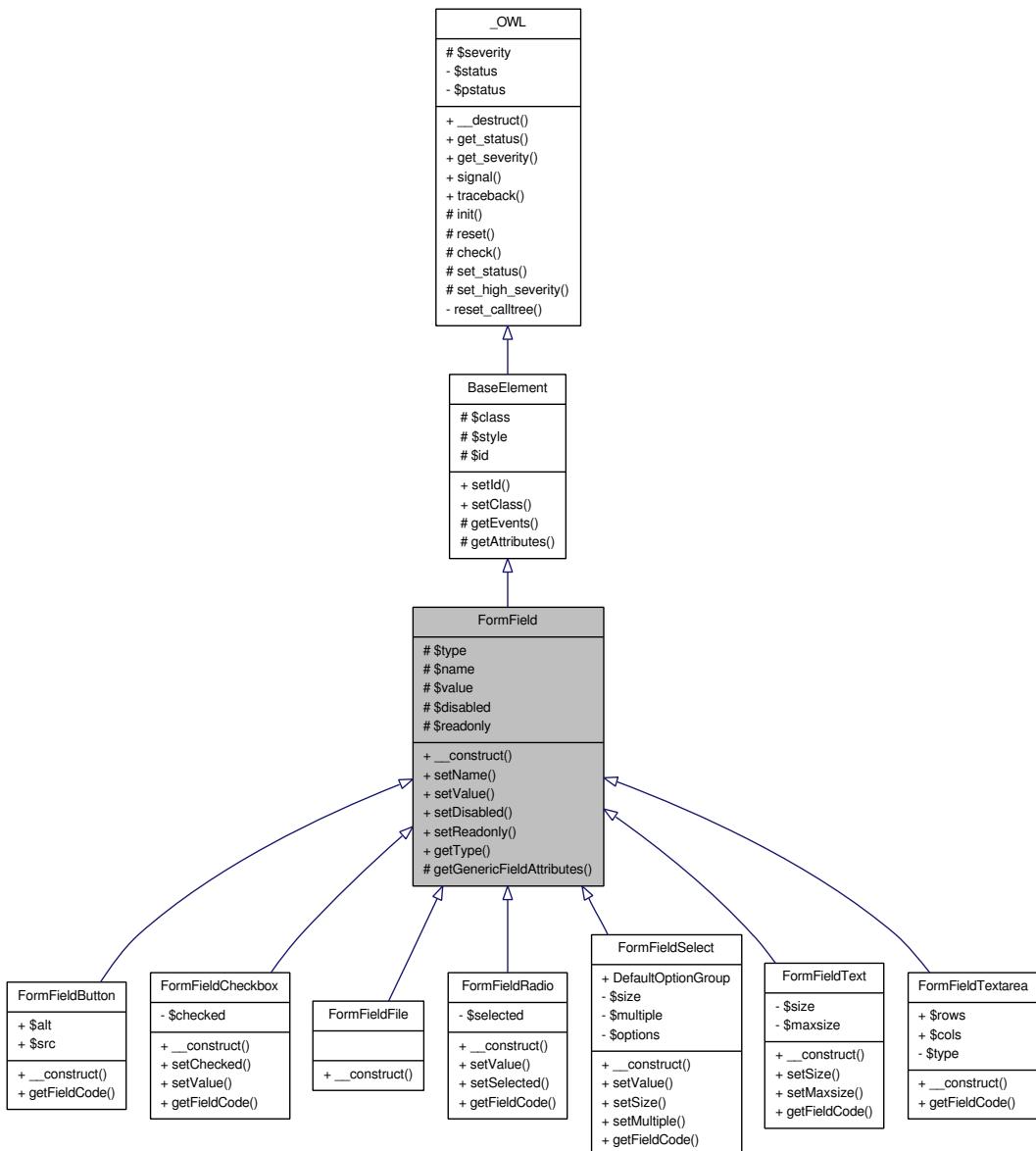
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/bo/[class.form.php](#)

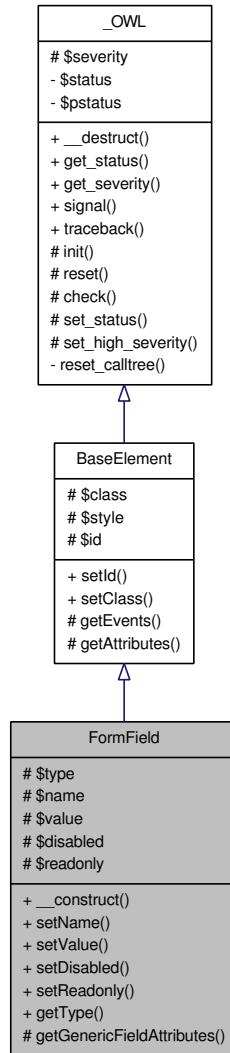
6.9 FormField Class Reference

Formfield.

Inheritance diagram for FormField:



Collaboration diagram for FormField:



Public Member Functions

- `__construct ()`
- `setName ($_value)`
- `setValue ($_value)`
- `setDisabled ($_value=true)`
- `setReadonly ($_value=true)`
- `getType ()`
- `setId ($_value)`
- `setClass ($_value)`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Protected Member Functions

- `getGenericFieldAttributes ($_ignore=array())`
- `getEvents ()`
- `getAttributes ()`
- `init ()`
- `reset ()`
- `check (&$object, $level)`
- `set_status ($status, $params=array())`
- `set_high_severity (&$object=null)`

Protected Attributes

- `$type`
- `$name`
- `$value`
- `$disabled`
- `$readonly`
- `$class = ''`
- `$style = ''`
- `$id = ''`
- `$severity`

6.9.1 Detailed Description

Formfield. Abstract base class for Formfield elements

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Oct 19, 2010 -- O van Eijk -- initial version

6.9.2 Constructor & Destructor Documentation

6.9.2.1 FormField::__construct ()

Class constructor;

Reimplemented in [FormFieldCheckbox](#), [FormFieldFile](#), [FormFieldRadio](#), [FormFieldSelect](#), and [FormFieldTextarea](#).

Referenced by [FormFieldText::__construct\(\)](#), and [FormFieldButton::__construct\(\)](#).

6.9.3 Member Function Documentation

6.9.3.1 __OWL::check (&\$ object, \$ level) [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

← *\$object* Pointer to an object to check against
← *\$level* The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.
Referenced by `SessionHandler::write()`.

6.9.3.2 _OWL::get_severity (\$ status = null) [inherited]

Get the current object severity level.

Parameters

← *\$status* An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References `_OWL::$status`.

Referenced by `LogHandler::compose_message()`, `LogHandler::log()`, and `DataHandler::set_key()`.

6.9.3.3 _OWL::get_status () [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by `SchemeHandler::compare()`.

6.9.3.4 BaseElement::getAttributes () [protected, inherited]

Return the general element attributes that are set

Returns

string attributes in HTML format (' fld="value" ...)

Referenced by `Form::openForm()`.

6.9.3.5 BaseElement::getEvents () [protected, inherited]

Referenced by `getGenericFieldAttributes()`.

6.9.3.6 FormField::getGenericFieldAttributes (\$ _ignore = array()) [protected]

Return the attributes for an HTML formfield in the " attrib='value' [...]" format

Parameters

$\leftarrow \$_ignore$ Array with attributes names that should be ignored, e.g. for a textarea, the value is not returned as an attribute.

Returns

Textstring with the HTML code

References BaseElement::getEvents().

Referenced by FormFieldTextarea::getFieldCode(), FormFieldText::getFieldCode(), FormFieldSelect::getFieldCode(), FormFieldRadio::getFieldCode(), FormFieldCheckbox::getFieldCode(), and FormFieldButton::getFieldCode().

6.9.3.7 FormField::getType ()

Give the field type

Returns

Field type

6.9.3.8 __OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References OWL::factory().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), LogHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), OWL::__construct(), Dispatcher::__construct(), and DataHandler::DataHandler().

6.9.3.9 __OWL::reset () [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References __OWL::reset_calltree().

Referenced by __OWL::check(), SessionHandler::read(), DataHandler::reset(), and __OWL::set_status().

6.9.3.10 __OWL::set_high_severity (&\$ object = null) [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.9.3.11 `_OWL::set_status ($status, $params = array ())` [final, protected, inherited]

Set the current object status to the specified value.

Parameters

$\leftarrow \$status$ OWL status code

$\leftarrow \$params$

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler::DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`, `Dispatcher::dispatch()`, `FormHandler::get()`, `DataHandler::get()`, `SchemeHandler::get_table_columns()`, `SchemeHandler::get_table_indexes()`, `UserHandler::login()`, `FileHandler::open()`, `DbHandler::open()`, `LogHandler::openLogFile()`, `DataHandler::prepare()`, `DbHandler::prepare_delete()`, `DbHandler::prepare_insert()`, `DbHandler::prepare_read()`, `DbHandler::prepare_update()`, `DbHandler::read()`, `FileHandler::read_line()`, `UserHandler::read_userdata()`, `DataHandler::reset()`, `SchemeHandler::scheme()`, `FormHandler::set()`, `DataHandler::set()`, `DataHandler::set_join()`, `DataHandler::set_key()`, `Form::setFieldAttributes()`, `FormFieldText::setMaxsize()`, `FormFieldRadio:: setSelected()`, `FormFieldText::setSize()`, `FormFieldSelect::setSize()`, `FormFieldSelect::setValue()`, `FormFieldRadio::setValue()`, `Form::showField()`, `SchemeHandler::table_description()`, `SchemeHandler::validate_scheme()`, and `DbHandler::write()`.

6.9.3.12 `BaseElement::setClass ($_value)` [inherited]

Set the element Class

Parameters

$\leftarrow \$_value$ Class name

6.9.3.13 `FormField::setDisabled ($_value = true)`

Set the Disabled boolean

Parameters

$\leftarrow \$_value$ Value indicating true (default) or false

6.9.3.14 `BaseElement::setId ($_value)` [inherited]

Set the element ID

Parameters

← *\$value* Identification

6.9.3.15 FormField::setName (\$value)

Set the field name

Parameters

← *\$value* Field name

6.9.3.16 FormField::set Readonly (\$value = true)

Set the Readonly boolean

Parameters

← *\$value* Value indicating true (default) or false

6.9.3.17 FormField::setValue (\$value)

Set the field value

Parameters

← *\$value* Field value

Reimplemented in [FormFieldCheckbox](#), [FormFieldRadio](#), and [FormFieldSelect](#).

6.9.3.18 _OWL::signal (\$level = OWL_INFO, &\$text = false) [inherited]

Display the message for the current object status

Parameters

← *\$level* An optional severity level; message will only be displayed when it is at least of this level.

\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References [ConfigHandler::get\(\)](#).

Referenced by [_OWL::set_status\(\)](#), and [_OWL::traceback\(\)](#).

6.9.3.19 _OWL::traceback (&\$ *text* = false, \$ *depth* = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References `_OWL::signal()`.

Referenced by `_OWL::check()`, `UserHandler::login()`, and `SessionHandler::read()`.

6.9.4 Member Data Documentation**6.9.4.1 BaseElement::\$class = ” [protected, inherited]**

Class specification

6.9.4.2 FormField::\$disabled [protected]

Boolean indicating a disabled field when true

6.9.4.3 BaseElement::\$id = ” [protected, inherited]

Element ID

6.9.4.4 FormField::\$name [protected]

Name of the formfield

6.9.4.5 FormField::\$readonly [protected]

Boolean indicating a readonly field when true

6.9.4.6 _OWL::\$severity [protected, inherited]

Severity level of the current object status

6.9.4.7 BaseElement::\$style = ” [protected, inherited]

Element style

6.9.4.8 **FormField::\$type** [protected]

Field type

Reimplemented in [FormFieldTextarea](#).

Referenced by [FormFieldText::__construct\(\)](#), and [FormFieldButton::__construct\(\)](#).

6.9.4.9 **FormField::\$value** [protected]

Field value

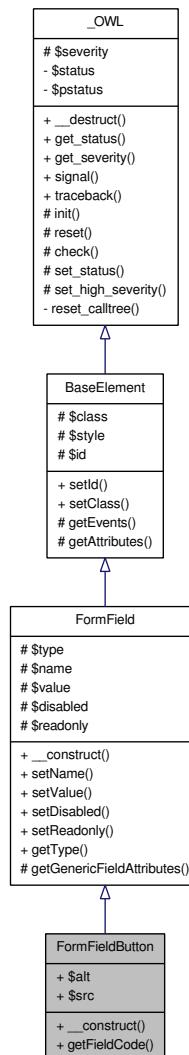
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/ui/[class.formfield.php](#)

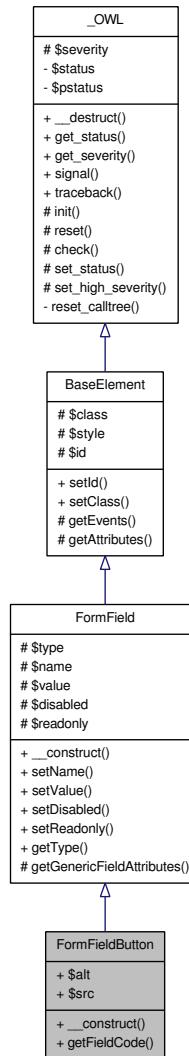
6.10 FormFieldButton Class Reference

Formfield.

Inheritance diagram for FormFieldButton:



Collaboration diagram for FormFieldButton:



Public Member Functions

- `__construct ($type= 'button')`
- `getFieldCode ()`
- `setName ($_value)`
- `setValue ($_value)`
- `setDisabled ($_value=true)`
- `setReadonly ($_value=true)`
- `getType ()`
- `setId ($_value)`
- `setClass ($_value)`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Public Attributes

- `$alt`
- `$src`

Protected Member Functions

- `getGenericFieldAttributes ($_ignore=array())`
- `getEvents ()`
- `getAttributes ()`
- `init ()`
- `reset ()`
- `check (&$object, $level)`
- `set _status ($status, $params=array())`
- `set _high _severity (&$object=null)`

Protected Attributes

- `$type`
- `$name`
- `$value`
- `$disabled`
- `$readonly`
- `$class = ''`
- `$style = ''`
- `$id = ''`
- `$severity`

6.10.1 Detailed Description

Formfield. Formfield button elements

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Oct 19, 2010 -- O van Eijk -- initial version

6.10.2 Constructor & Destructor Documentation

6.10.2.1 FormFieldButton::__construct (\$ type = 'button')

Class constructor;

Parameters

← `$type` Element type: button (default), image, submit or reset

References `FormField::$type`, and `FormField::__construct()`.

6.10.3 Member Function Documentation

6.10.3.1 `_OWL::check (&$ object, $ level)` [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← `$object` Pointer to an object to check against
- ← `$level` The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.10.3.2 `_OWL::get_severity ($ status = null)` [inherited]

Get the current object severity level.

Parameters

- ← `$status` An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References `_OWL::$status`.

Referenced by `LogHandler::compose_message()`, `LogHandler::log()`, and `DataHandler::set_key()`.

6.10.3.3 `_OWL::get_status ()` [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by `SchemeHandler::compare()`.

6.10.3.4 `BaseElement::getAttributes ()` [protected, inherited]

Return the general element attributes that are set

Returns

string attributes in HTML format (' fld="value"...')

Referenced by `Form::openForm()`.

6.10.3.5 BaseElement::getEvents () [protected, inherited]

Referenced by FormField::getGenericFieldAttributes().

6.10.3.6 FormFieldButton::getFieldCode ()

Return the HTML code to display the form element

Returns

Textstring with the complete code for the form element

References FormField::getGenericFieldAttributes().

6.10.3.7 FormField::getGenericFieldAttributes (\$ _ignore = array()) [protected, inherited]

Return the attributes for an HTML formfield in the " attrib='value' [...]" format

Parameters

← *\$ _ignore* Array with attributes names that should be ignored, e.g. for a textarea, the value is not returned as an attribute.

Returns

Textstring with the HTML code

References BaseElement::getEvents().

Referenced by FormFieldTextarea::getFieldCode(), FormFieldText::getFieldCode(), FormFieldSelect::getFieldCode(), FormFieldRadio::getFieldCode(), FormFieldCheckbox::getFieldCode(), and getFieldCode().

6.10.3.8 FormField::getType () [inherited]

Give the field type

Returns

Field type

6.10.3.9 _OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References OWL::factory().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), LogHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), OWL::__construct(), Dispatcher::__construct(), and DataHandler::DataHandler().

6.10.3.10 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References `_OWL::reset_calltree()`.

Referenced by `_OWL::check()`, `SessionHandler::read()`, `DataHandler::reset()`, and `_OWL::set_status()`.

6.10.3.11 `_OWL::set_high_severity (&$ object = null)` [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.10.3.12 `_OWL::set_status ($ status, $ params = array ())` [final, protected, inherited]

Set the current object status to the specified value.

Parameters

$\leftarrow \$status$ [OWL](#) status code

$\leftarrow \$params$

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler::DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`, `Dispatcher::dispatch()`, `FormHandler::get()`, `DataHandler::get()`, `SchemeHandler::get_table_columns()`, `SchemeHandler::get_table_indexes()`, `UserHandler::login()`, `FileHandler::open()`, `DbHandler::open()`, `LogHandler::open_logfile()`, `DataHandler::prepare()`, `DbHandler::prepare_delete()`, `DbHandler::prepare_insert()`, `DbHandler::prepare_read()`, `DbHandler::prepare_update()`, `DbHandler::read()`, `FileHandler::read_line()`, `UserHandler::read_userdata()`, `DataHandler::reset()`, `SchemeHandler::scheme()`, `FormHandler::set()`, `DataHandler::set()`, `DataHandler::set_join()`, `DataHandler::set_key()`, `Form::setFieldAttributes()`, `FormFieldText::setMaxsize()`, `FormFieldRadio::setSelected()`, `FormFieldText::setSize()`, `FormFieldSelect::setSize()`, `FormFieldSelect::setValue()`, `FormFieldRadio::setValue()`, `Form::showField()`, `SchemeHandler::table_description()`, `SchemeHandler::validate_scheme()`, and `DbHandler::write()`.

6.10.3.13 `BaseElement::setClass ($ _value)` [inherited]

Set the element Class

Parameters

$\leftarrow \$_value$ Class name

6.10.3.14 FormField::setDisabled (\$ _value = true) [inherited]

Set the Disabled boolean

Parameters

← *\$ _value* Value indicating true (default) or false

6.10.3.15 BaseElement::setId (\$ _value) [inherited]

Set the element ID

Parameters

← *\$ _value* Identification

6.10.3.16 FormField::setName (\$ _value) [inherited]

Set the field name

Parameters

← *\$ _value* Field name

6.10.3.17 FormField::set Readonly (\$ _value = true) [inherited]

Set the Readonly boolean

Parameters

← *\$ _value* Value indicating true (default) or false

6.10.3.18 FormField::setValue (\$ _value) [inherited]

Set the field value

Parameters

← *\$ _value* Field value

Reimplemented in [FormFieldCheckbox](#), [FormFieldRadio](#), and [FormFieldSelect](#).

6.10.3.19 _OWL::signal (\$ level = OWL_INFO, &\$ text = false) [inherited]

Display the message for the current object status

Parameters

← *\$level* An optional severity level; message will only be displayed when it is at least of this level.

\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by _OWL::set_status(), and _OWL::traceback().

6.10.3.20 _OWL::traceback (&\$ *text* = false, \$ *depth* = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References _OWL::signal().

Referenced by _OWL::check(), UserHandler::login(), and SessionHandler::read().

6.10.4 Member Data Documentation**6.10.4.1 FormFieldButton::\$alt**

Alternate text (image-type only)

6.10.4.2 BaseElement::\$class = " " [protected, inherited]

Class specification

6.10.4.3 FormField::\$disabled [protected, inherited]

Boolean indicating a disabled field when true

6.10.4.4 BaseElement::\$id = " " [protected, inherited]

Element ID

6.10.4.5 FormField::\$name [protected, inherited]

Name of the formfield

6.10.4.6 FormField::\$readonly [protected, inherited]

Boolean indicating a readonly field when true

6.10.4.7 __OWL::\$severity [protected, inherited]

Severity level of the current object status

6.10.4.8 FormFieldButton::\$src

Image src (image-type only)

6.10.4.9 BaseElement::\$style = "" [protected, inherited]

Element style

6.10.4.10 FormField::\$type [protected, inherited]

Field type

Reimplemented in [FormFieldTextarea](#).

Referenced by FormFieldText::__construct(), and __construct().

6.10.4.11 FormField::\$value [protected, inherited]

Field value

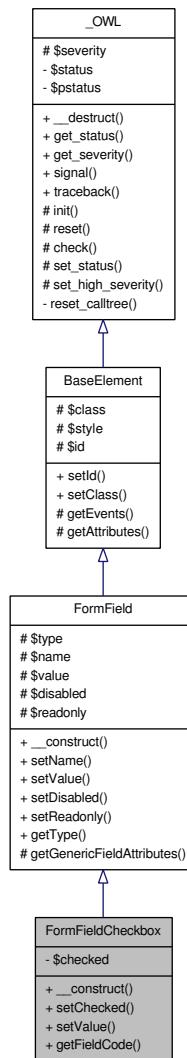
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/ui/formfields/[class.formfield.button.php](#)

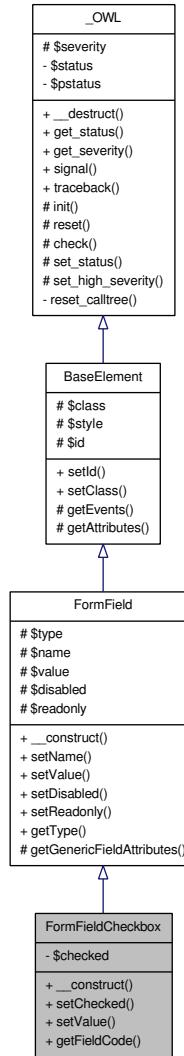
6.11 FormFieldCheckbox Class Reference

Formfield.

Inheritance diagram for FormFieldCheckbox:



Collaboration diagram for FormFieldCheckbox:



Public Member Functions

- `__construct ()`
- `setChecked ($_value=true)`
- `setValue ($_value)`
- `getFieldCode ()`
- `setName ($_value)`
- `setDisabled ($_value=true)`
- `setReadonly ($_value=true)`
- `getType ()`
- `setId ($_value)`
- `setClass ($_value)`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Protected Member Functions

- [getGenericFieldAttributes](#) (\$_ignore=array())
- [getEvents](#) ()
- [getAttributes](#) ()
- [init](#) ()
- [reset](#) ()
- [check](#) (&\$object, \$level)
- [set_status](#) (\$status, \$params=array())
- [set_high_severity](#) (&\$object=null)

Protected Attributes

- [\\$type](#)
- [\\$name](#)
- [\\$value](#)
- [\\$disabled](#)
- [\\$readonly](#)
- [\\$class](#) = ''
- [\\$style](#) = ''
- [\\$id](#) = ''
- [\\$severity](#)

Private Attributes

- [\\$checked](#)

6.11.1 Detailed Description

Formfield. Formfield Checkbox elements

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Oct 19, 2010 -- O van Eijk -- initial version

6.11.2 Constructor & Destructor Documentation

6.11.2.1 FormFieldCheckbox::__construct ()

Class constructor;

Reimplemented from [FormField](#).

6.11.3 Member Function Documentation

6.11.3.1 `_OWL::check (&$ object, $ level)` [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← `$object` Pointer to an object to check against
- ← `$level` The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.11.3.2 `_OWL::get_severity ($ status = null)` [inherited]

Get the current object severity level.

Parameters

- ← `$status` An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References `_OWL::$status`.

Referenced by `LogHandler::compose_message()`, `LogHandler::log()`, and `DataHandler::set_key()`.

6.11.3.3 `_OWL::get_status ()` [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by `SchemeHandler::compare()`.

6.11.3.4 `BaseElement::getAttributes ()` [protected, inherited]

Return the general element attributes that are set

Returns

string attributes in HTML format (' fld="value"...')

Referenced by `Form::openForm()`.

6.11.3.5 BaseElement::getEvents () [protected, inherited]

Referenced by FormField::getGenericFieldAttributes().

6.11.3.6 FormFieldCheckbox::getFieldCode ()

Return the HTML code to display the form element

Returns

Textstring with the complete code for the form element

References FormField::getGenericFieldAttributes().

6.11.3.7 FormField::getGenericFieldAttributes (\$ _ignore = array()) [protected, inherited]

Return the attributes for an HTML formfield in the " attrib='value' [...]" format

Parameters

← *\$ _ignore* Array with attributes names that should be ignored, e.g. for a textarea, the value is not returned as an attribute.

Returns

Textstring with the HTML code

References BaseElement::getEvents().

Referenced by FormFieldTextarea::getFieldCode(), FormFieldText::getFieldCode(), FormFieldSelect::getFieldCode(), FormFieldRadio::getFieldCode(), getFieldCode(), and FormFieldButton::getFieldCode().

6.11.3.8 FormField::getType () [inherited]

Give the field type

Returns

Field type

6.11.3.9 _OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References OWL::factory().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), LogHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), OWL::__construct(), Dispatcher::__construct(), and DataHandler::DataHandler().

6.11.3.10 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References `_OWL::reset_calltree()`.

Referenced by `_OWL::check()`, `SessionHandler::read()`, `DataHandler::reset()`, and `_OWL::set_status()`.

6.11.3.11 `_OWL::set_high_severity (&$ object = null)` [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.11.3.12 `_OWL::set_status ($status, $params = array ())` [final, protected, inherited]

Set the current object status to the specified value.

Parameters

$\leftarrow \$status$ [OWL](#) status code

$\leftarrow \$params$

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler::DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`, `Dispatcher::dispatch()`, `FormHandler::get()`, `DataHandler::get()`, `SchemeHandler::get_table_columns()`, `SchemeHandler::get_table_indexes()`, `UserHandler::login()`, `FileHandler::open()`, `DbHandler::open()`, `LogHandler::open_logfile()`, `DataHandler::prepare()`, `DbHandler::prepare_delete()`, `DbHandler::prepare_insert()`, `DbHandler::prepare_read()`, `DbHandler::prepare_update()`, `DbHandler::read()`, `FileHandler::read_line()`, `UserHandler::read_userdata()`, `DataHandler::reset()`, `SchemeHandler::scheme()`, `FormHandler::set()`, `DataHandler::set()`, `DataHandler::set_join()`, `DataHandler::set_key()`, `Form::setFieldAttributes()`, `FormFieldText::setMaxsize()`, `FormFieldRadio::setSelected()`, `FormFieldText::setSize()`, `FormFieldSelect::setSize()`, `FormFieldSelect::setValue()`, `FormFieldRadio::set_value()`, `Form::showField()`, `SchemeHandler::table_description()`, `SchemeHandler::validate_scheme()`, and `DbHandler::write()`.

6.11.3.13 `FormFieldCheckbox::setChecked ($_value = true)`

Set the Checked boolean

Parameters

$\leftarrow \$_value$ Value indicating true (default) or false

6.11.3.14 BaseElement::setClass (\$ _ value) [inherited]

Set the element Class

Parameters

← *\$ _ value* Class name

6.11.3.15 FormField::setDisabled (\$ _ value = true) [inherited]

Set the Disabled boolean

Parameters

← *\$ _ value* Value indicating true (default) or false

6.11.3.16 BaseElement::setId (\$ _ value) [inherited]

Set the element ID

Parameters

← *\$ _ value* Identification

6.11.3.17 FormField::setName (\$ _ value) [inherited]

Set the field name

Parameters

← *\$ _ value* Field name

6.11.3.18 FormField::set Readonly (\$ _ value = true) [inherited]

Set the Readonly boolean

Parameters

← *\$ _ value* Value indicating true (default) or false

6.11.3.19 FormFieldCheckbox::setValue (\$ _ value)

Reimplement; value defaults to 1 for checkboxes

Parameters

← *\$ _ value* Field value

Reimplemented from [FormField](#).

6.11.3.20 _OWL::signal (\$ level = OWL_INFO, &\$ text = false) [inherited]

Display the message for the current object status

Parameters

← **\$level** An optional severity level; message will only be displayed when it is at least of this level.

\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by _OWL::set_status(), and _OWL::traceback().

6.11.3.21 _OWL::traceback (&\$ text = false, \$ depth = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References _OWL::signal().

Referenced by _OWL::check(), UserHandler::login(), and SessionHandler::read().

6.11.4 Member Data Documentation

6.11.4.1 FormFieldCheckbox::\$checked [private]

Boolean set to true when the box is checked

6.11.4.2 BaseElement::\$class = " " [protected, inherited]

Class specification

6.11.4.3 FormField::\$disabled [protected, inherited]

Boolean indicating a disabled field when true

6.11.4.4 BaseElement::\$id = ” [protected, inherited]

Element ID

6.11.4.5 FormField::\$name [protected, inherited]

Name of the formfield

6.11.4.6 FormField::\$readonly [protected, inherited]

Boolean indicating a readonly field when true

6.11.4.7 _OWL::\$severity [protected, inherited]

Severity level of the current object status

6.11.4.8 BaseElement::\$style = ” [protected, inherited]

Element style

6.11.4.9 FormField::\$type [protected, inherited]

Field type

Reimplemented in [FormFieldTextarea](#).

Referenced by FormFieldText::__construct(), and FormFieldButton::__construct().

6.11.4.10 FormField::\$value [protected, inherited]

Field value

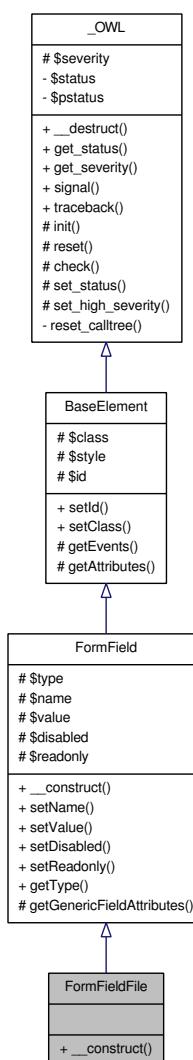
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/ui/formfields/[class.formfield.checkbox.php](#)

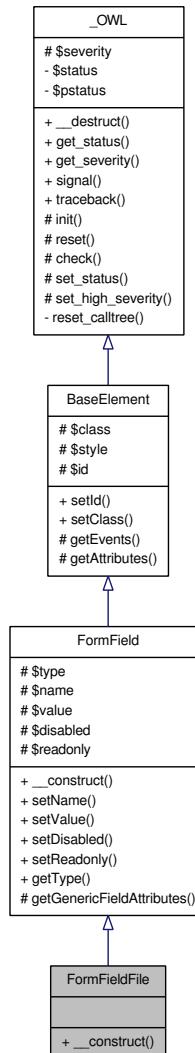
6.12 FormFieldFile Class Reference

Formfield.

Inheritance diagram for FormFieldFile:



Collaboration diagram for FormFieldFile:



Public Member Functions

- `__construct ()`
- `setName ($_value)`
- `setValue ($_value)`
- `setDisabled ($_value=true)`
- `setReadonly ($_value=true)`
- `getType ()`
- `setId ($_value)`
- `setClass ($_value)`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Protected Member Functions

- `getGenericFieldAttributes` (`$_ignore=array()`)
- `getEvents` ()
- `getAttributes` ()
- `init` ()
- `reset` ()
- `check` (`&$object, $level`)
- `set_status` (`$status, $params=array()`)
- `set_high_severity` (`&$object=null`)

Protected Attributes

- `$type`
- `$name`
- `$value`
- `$disabled`
- `$readonly`
- `$class = ''`
- `$style = ''`
- `$id = ''`
- `$severity`

6.12.1 Detailed Description

Formfield. Formfield File input elements

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Oct 19, 2010 -- O van Eijk -- initial version

6.12.2 Constructor & Destructor Documentation

6.12.2.1 FormFieldFile::__construct ()

Class constructor;

Reimplemented from [FormField](#).

6.12.3 Member Function Documentation

6.12.3.1 __OWL::check (&\$ object, \$ level) [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← *\$object* Pointer to an object to check against
- ← *\$level* The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.12.3.2 _OWL::get_severity (\$ status = null) [inherited]

Get the current object severity level.

Parameters

- ← *\$status* An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References `_OWL::$status`.

Referenced by `LogHandler::compose_message()`, `LogHandler::log()`, and `DataHandler::set_key()`.

6.12.3.3 _OWL::get_status () [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by `SchemeHandler::compare()`.

6.12.3.4 BaseElement::getAttributes () [protected, inherited]

Return the general element attributes that are set

Returns

string attributes in HTML format (' fld="value" ...)

Referenced by `Form::openForm()`.

6.12.3.5 BaseElement::getEvents () [protected, inherited]

Referenced by `FormField::getGenericFieldAttributes()`.

6.12.3.6 FormField::getGenericFieldAttributes (\$ _ignore = array()) [protected, inherited]

Return the attributes for an HTML formfield in the " attrib='value' [...]" format

Parameters

← **\$ _ignore** Array with attributes names that should be ignored, e.g. for a textarea, the value is not returned as an attribute.

Returns

Textstring with the HTML code

References BaseElement::getEvents().

Referenced by FormFieldTextarea::getFieldCode(), FormFieldText::getFieldCode(), FormFieldSelect::getFieldCode(), FormFieldRadio::getFieldCode(), FormFieldCheckbox::getFieldCode(), and FormFieldButton::getFieldCode().

6.12.3.7 FormField::getType () [inherited]

Give the field type

Returns

Field type

6.12.3.8 _OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References OWL::factory().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), LogHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), OWL::__construct(), Dispatcher::__construct(), and DataHandler::DataHandler().

6.12.3.9 _OWL::reset () [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References _OWL::reset_calltree().

Referenced by _OWL::check(), SessionHandler::read(), DataHandler::reset(), and _OWL::set_status().

6.12.3.10 _OWL::set_high_severity (&\$ object = null) [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.12.3.11 `_OWL::set_status ($status, $params = array ()) [final, protected, inherited]`

Set the current object status to the specified value.

Parameters

$\leftarrow \$status$ OWL status code

$\leftarrow \$params$

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler::DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`, `Dispatcher::dispatch()`, `FormHandler::get()`, `DataHandler::get()`, `SchemeHandler::get_table_columns()`, `SchemeHandler::get_table_indexes()`, `UserHandler::login()`, `FileHandler::open()`, `DbHandler::open()`, `LogHandler::openLogFile()`, `DataHandler::prepare()`, `DbHandler::prepare_delete()`, `DbHandler::prepare_insert()`, `DbHandler::prepare_read()`, `DbHandler::prepare_update()`, `DbHandler::read()`, `FileHandler::read_line()`, `UserHandler::read_userdata()`, `DataHandler::reset()`, `SchemeHandler::scheme()`, `FormHandler::set()`, `DataHandler::set()`, `DataHandler::set_join()`, `DataHandler::set_key()`, `Form::setFieldAttributes()`, `FormFieldText::setMaxsize()`, `FormFieldRadio:: setSelected()`, `FormFieldText::setSize()`, `FormFieldSelect::setSize()`, `FormFieldSelect::setValue()`, `FormFieldRadio::setValue()`, `Form::showField()`, `SchemeHandler::table_description()`, `SchemeHandler::validate_scheme()`, and `DbHandler::write()`.

6.12.3.12 `BaseElement::setClass ($_value) [inherited]`

Set the element Class

Parameters

$\leftarrow \$_value$ Class name

6.12.3.13 `FormField::setDisabled ($_value = true) [inherited]`

Set the Disabled boolean

Parameters

$\leftarrow \$_value$ Value indicating true (default) or false

6.12.3.14 `BaseElement::setId ($_value) [inherited]`

Set the element ID

Parameters

← *\$value* Identification

6.12.3.15 FormField::setName (\$value) [inherited]

Set the field name

Parameters

← *\$value* Field name

6.12.3.16 FormField::set Readonly (\$value = true) [inherited]

Set the Readonly boolean

Parameters

← *\$value* Value indicating true (default) or false

6.12.3.17 FormField::setValue (\$value) [inherited]

Set the field value

Parameters

← *\$value* Field value

Reimplemented in [FormFieldCheckbox](#), [FormFieldRadio](#), and [FormFieldSelect](#).

6.12.3.18 _OWL::signal (\$level = OWL_INFO, &\$text = false) [inherited]

Display the message for the current object status

Parameters

← *\$level* An optional severity level; message will only be displayed when it is at least of this level.

\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References [ConfigHandler::get\(\)](#).

Referenced by [_OWL::set_status\(\)](#), and [_OWL::traceback\(\)](#).

6.12.3.19 `_OWL::traceback (&$ text = false, $ depth = 0)` [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

`$text` Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← `$depth` This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References `_OWL::signal()`.

Referenced by `_OWL::check()`, `UserHandler::login()`, and `SessionHandler::read()`.

6.12.4 Member Data Documentation

6.12.4.1 `BaseElement::$class = ”` [protected, inherited]

Class specification

6.12.4.2 `FormField::$disabled` [protected, inherited]

Boolean indicating a disabled field when true

6.12.4.3 `BaseElement::$id = ”` [protected, inherited]

Element ID

6.12.4.4 `FormField::$name` [protected, inherited]

Name of the formfield

6.12.4.5 `FormField::$readonly` [protected, inherited]

Boolean indicating a readonly field when true

6.12.4.6 `_OWL::$severity` [protected, inherited]

Severity level of the current object status

6.12.4.7 `BaseElement::$style = ”` [protected, inherited]

Element style

6.12.4.8 FormField::\$type [protected, inherited]

Field type

Reimplemented in [FormFieldTextarea](#).

Referenced by FormFieldText::__construct(), and FormFieldButton::__construct().

6.12.4.9 FormField::\$value [protected, inherited]

Field value

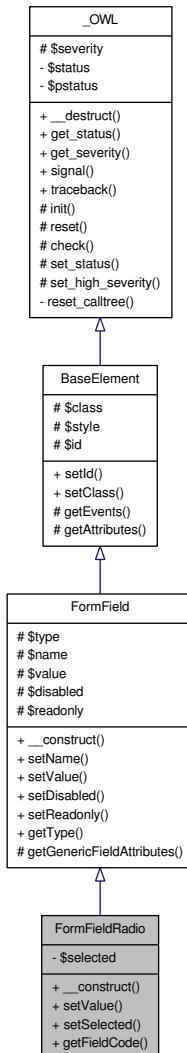
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/ui/formfields/[class.formfield.file.php](#)

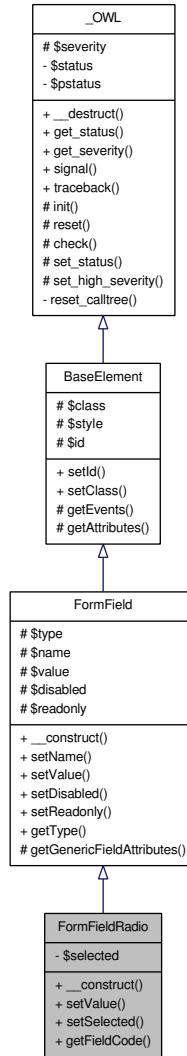
6.13 FormFieldRadio Class Reference

Formfield.

Inheritance diagram for FormFieldRadio:



Collaboration diagram for FormFieldRadio:



Public Member Functions

- `__construct ()`
- `setValue ($_value)`
- `setSelected ($_value)`
- `getFieldCode ()`
- `setName ($_value)`
- `setDisabled ($_value=true)`
- `setReadonly ($_value=true)`
- `getType ()`
- `setId ($_value)`
- `setClass ($_value)`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Protected Member Functions

- [getGenericFieldAttributes](#) (\$_ignore=array())
- [getEvents](#) ()
- [getAttributes](#) ()
- [init](#) ()
- [reset](#) ()
- [check](#) (&\$object, \$level)
- [set_status](#) (\$status, \$params=array())
- [set_high_severity](#) (&\$object=null)

Protected Attributes

- [\\$type](#)
- [\\$name](#)
- [\\$value](#)
- [\\$disabled](#)
- [\\$readonly](#)
- [\\$class](#) = ''
- [\\$style](#) = ''
- [\\$id](#) = ''
- [\\$severity](#)

Private Attributes

- [\\$selected](#)

6.13.1 Detailed Description

Formfield. Formfield radio elements

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Oct 19, 2010 -- O van Eijk -- initial version

6.13.2 Constructor & Destructor Documentation

6.13.2.1 FormFieldRadio::__construct ()

Class constructor;

Reimplemented from [FormField](#).

6.13.3 Member Function Documentation

6.13.3.1 `_OWL::check (&$ object, $ level)` [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← `$object` Pointer to an object to check against
- ← `$level` The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.13.3.2 `_OWL::get_severity ($ status = null)` [inherited]

Get the current object severity level.

Parameters

- ← `$status` An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References `_OWL::$status`.

Referenced by `LogHandler::compose_message()`, `LogHandler::log()`, and `DataHandler::set_key()`.

6.13.3.3 `_OWL::get_status ()` [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by `SchemeHandler::compare()`.

6.13.3.4 `BaseElement::getAttributes ()` [protected, inherited]

Return the general element attributes that are set

Returns

string attributes in HTML format (' fld="value"...')

Referenced by `Form::openForm()`.

6.13.3.5 BaseElement::getEvents () [protected, inherited]

Referenced by FormField::getGenericFieldAttributes().

6.13.3.6 FormFieldRadio::getFieldCode ()

Return the HTML code to display the form elements

Returns

Array with textstrings for each radio button in this this set.

References FormField::getGenericFieldAttributes().

6.13.3.7 FormField::getGenericFieldAttributes (\$ _ignore = array()) [protected, inherited]

Return the attributes for an HTML formfield in the " attrib='value' [...]" format

Parameters

← *\$ _ignore* Array with attributes names that should be ignored, e.g. for a textarea, the value is not returned as an attribute.

Returns

Textstring with the HTML code

References BaseElement::getEvents().

Referenced by FormFieldTextarea::getFieldCode(), FormFieldText::getFieldCode(), FormFieldSelect::getFieldCode(), getFieldCode(), FormFieldCheckbox::getFieldCode(), and FormFieldButton::getFieldCode().

6.13.3.8 FormField::getType () [inherited]

Give the field type

Returns

Field type

6.13.3.9 _OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References OWL::factory().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), LogHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), OWL::__construct(), Dispatcher::__construct(), and DataHandler::DataHandler().

6.13.3.10 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References `_OWL::reset_calltree()`.

Referenced by `_OWL::check()`, `SessionHandler::read()`, `DataHandler::reset()`, and `_OWL::set_status()`.

6.13.3.11 `_OWL::set_high_severity (&$ object = null)` [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.13.3.12 `_OWL::set_status ($ status, $ params = array ())` [final, protected, inherited]

Set the current object status to the specified value.

Parameters

$\leftarrow \$status$ [OWL](#) status code
 $\leftarrow \$params$

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler::DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`, `Dispatcher::dispatch()`, `FormHandler::get()`, `DataHandler::get()`, `SchemeHandler::get_table_columns()`, `SchemeHandler::get_table_indexes()`, `UserHandler::login()`, `FileHandler::open()`, `DbHandler::open()`, `LogHandler::open_logfile()`, `DataHandler::prepare()`, `DbHandler::prepare_delete()`, `DbHandler::prepare_insert()`, `DbHandler::prepare_read()`, `DbHandler::prepare_update()`, `DbHandler::read()`, `FileHandler::read_line()`, `UserHandler::read_userdata()`, `DataHandler::reset()`, `SchemeHandler::scheme()`, `FormHandler::set()`, `DataHandler::set()`, `DataHandler::set_join()`, `DataHandler::set_key()`, `Form::setFieldAttributes()`, `FormFieldText::setMaxsize()`, `setSelected()`, `FormFieldText::setSize()`, `FormFieldSelect::setSize()`, `FormFieldSelect::setValue()`, `setValue()`, `Form::showField()`, `SchemeHandler::table_description()`, `SchemeHandler::validate_scheme()`, and `DbHandler::write()`.

6.13.3.13 `BaseElement::setClass ($ _value)` [inherited]

Set the element Class

Parameters

$\leftarrow \$_value$ Class name

6.13.3.14 FormField::setDisabled (\$ _value = true) [inherited]

Set the Disabled boolean

Parameters

← *\$ _value* Value indicating true (default) or false

6.13.3.15 BaseElement::setId (\$ _value) [inherited]

Set the element ID

Parameters

← *\$ _value* Identification

6.13.3.16 FormField::setName (\$ _value) [inherited]

Set the field name

Parameters

← *\$ _value* Field name

6.13.3.17 FormField::set Readonly (\$ _value = true) [inherited]

Set the Readonly boolean

Parameters

← *\$ _value* Value indicating true (default) or false

6.13.3.18 FormFieldRadio::setSelected (\$ _value)

Define the selected value

Parameters

← *\$ _value* Preselected value

References *_OWL::set_status()*.

6.13.3.19 FormFieldRadio::setValue (\$ _value)

Reimplement, multiple values are supported

Parameters

← *\$ _value* Field value

Reimplemented from [FormField](#).

References *_OWL::set_status()*.

6.13.3.20 _OWL::signal (\$ level = OWL_INFO, &\$ text = false) [inherited]

Display the message for the current object status

Parameters

← **\$level** An optional severity level; message will only be displayed when it is at least of this level.

\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by _OWL::set_status(), and _OWL::traceback().

6.13.3.21 _OWL::traceback (&\$ text = false, \$ depth = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References _OWL::signal().

Referenced by _OWL::check(), UserHandler::login(), and SessionHandler::read().

6.13.4 Member Data Documentation

6.13.4.1 BaseElement::\$class = "" [protected, inherited]

Class specification

6.13.4.2 FormField::\$disabled [protected, inherited]

Boolean indicating a disabled field when true

6.13.4.3 BaseElement::\$id = "" [protected, inherited]

Element ID

6.13.4.4 FormField::\$name [protected, inherited]

Name of the formfield

6.13.4.5 FormField::\$readonly [protected, inherited]

Boolean indicating a readonly field when true

6.13.4.6 FormFieldRadio::\$selected [private]

Holds the value that must be preselected

6.13.4.7 _OWL::\$severity [protected, inherited]

Severity level of the current object status

6.13.4.8 BaseElement::\$style = "" [protected, inherited]

Element style

6.13.4.9 FormField::\$type [protected, inherited]

Field type

Reimplemented in [FormFieldTextarea](#).

Referenced by [FormFieldText::__construct\(\)](#), and [FormFieldButton::__construct\(\)](#).

6.13.4.10 FormField::\$value [protected, inherited]

Field value

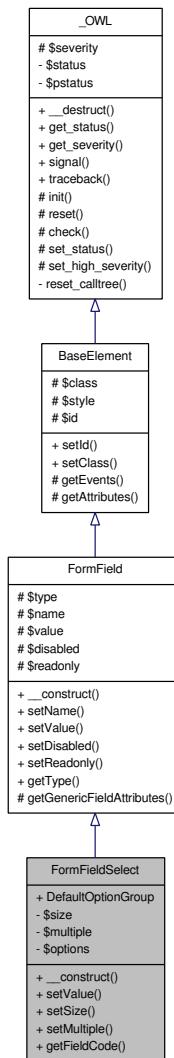
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/ui/formfields/[class.formfield.radio.php](#)

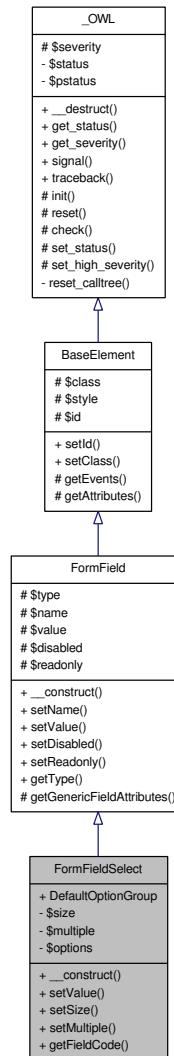
6.14 FormFieldSelect Class Reference

Formfield.

Inheritance diagram for FormFieldSelect:



Collaboration diagram for FormFieldSelect:



Public Member Functions

- `__construct ()`
- `setValue ($_value)`
- `setSize ($size)`
- `setMultiple ($_value=true)`
- `getFieldCode ()`
- `setName ($_value)`
- `setDisabled ($_value=true)`
- `setReadonly ($_value=true)`
- `getType ()`
- `setId ($_value)`
- `setClass ($_value)`
- `get_status ()`
- `get_severity ($status=null)`

- `signal` (\$level=OWL_INFO, &\$text=false)
- `traceback` (&\$text=false, \$depth=0)

Public Attributes

- const `DefaultOptionGroup` = '_OWL_OptGroup_'

Protected Member Functions

- `getGenericFieldAttributes` (\$_ignore=array())
- `getEvents` ()
- `getAttributes` ()
- `init` ()
- `reset` ()
- `check` (&\$object, \$level)
- `set_status` (\$status, \$params=array())
- `set_high_severity` (&\$object=null)

Protected Attributes

- `$type`
- `$name`
- `$value`
- `$disabled`
- `$readonly`
- `$class = ''`
- `$style = ''`
- `$id = ''`
- `$severity`

Private Attributes

- `$size`
- `$multiple`
- `$options`

6.14.1 Detailed Description

Formfield. Formfield selectlist elements

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Oct 19, 2010 -- O van Eijk -- initial version

6.14.2 Constructor & Destructor Documentation

6.14.2.1 `FormFieldSelect::__construct ()`

Class constructor;

Reimplemented from [FormField](#).

6.14.3 Member Function Documentation

6.14.3.1 `_OWL::check (&$ object, $ level) [protected, inherited]`

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← `$object` Pointer to an object to check against
- ← `$level` The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.14.3.2 `_OWL::get_severity ($ status = null) [inherited]`

Get the current object severity level.

Parameters

- ← `$status` An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References `_OWL::$status`.

Referenced by `LogHandler::compose_message()`, `LogHandler::log()`, and `DataHandler::set_key()`.

6.14.3.3 `_OWL::get_status () [final, inherited]`

Get the current object status.

Returns

Object's status code

Referenced by `SchemeHandler::compare()`.

6.14.3.4 BaseElement::getAttributes () [protected, inherited]

Return the general element attributes that are set

Returns

String attributes in HTML format (' fld="value"...')

Referenced by Form::openForm().

6.14.3.5 BaseElement::getEvents () [protected, inherited]

Referenced by FormField::getGenericFieldAttributes().

6.14.3.6 FormFieldSelect::getFieldCode ()

Return the HTML code to display the form elements

Returns

String with the element formcode

References FormField::getGenericFieldAttributes().

6.14.3.7 FormField::getGenericFieldAttributes (\$ _ignore = array()) [protected, inherited]

Return the attributes for an HTML formfield in the " attrib='value' [...]" format

Parameters

← *\$ _ignore* Array with attributes names that should be ignored, e.g. for a textarea, the value is not returned as an attribute.

Returns

Textstring with the HTML code

References BaseElement::getEvents().

Referenced by FormFieldTextarea::getFieldCode(), FormFieldText::getFieldCode(), getFieldCode(), FormFieldRadio::getFieldCode(), FormFieldCheckbox::getFieldCode(), and FormFieldButton::getFieldCode().

6.14.3.8 FormField::getType () [inherited]

Give the field type

Returns

Field type

6.14.3.9 `_OWL::init ()` [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References `OWL::factory()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `LogHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `OWL::__construct()`, `Dispatcher::__construct()`, and `DataHandler::DataHandler()`.

6.14.3.10 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in `DbHandler`, and `SchemeHandler`.

References `_OWL::reset_calltree()`.

Referenced by `_OWL::check()`, `SessionHandler::read()`, `DataHandler::reset()`, and `_OWL::set_status()`.

6.14.3.11 `_OWL::set_high_severity (&$ object = null)` [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.14.3.12 `_OWL::set_status ($status, $params = array ())` [final, protected, inherited]

Set the current object status to the specified value.

Parameters

$\leftarrow \$status$ `OWL` status code

$\leftarrow \$params$

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler::DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`, `Dispatcher::dispatch()`, `FormHandler::get()`, `DataHandler::get()`, `SchemeHandler::get_table_columns()`, `SchemeHandler::get_table_indexes()`, `UserHandler::login()`, `FileHandler::open()`, `DbHandler::open()`, `LogHandler::open_logfile()`, `DataHandler::prepare()`, `DbHandler::prepare_delete()`, `DbHandler::prepare_insert()`, `DbHandler::prepare_read()`, `DbHandler::prepare_update()`, `DbHandler::read()`, `FileHandler::read_line()`, `UserHandler::read_userdata()`,

DataHandler::reset(), SchemeHandler::scheme(), FormHandler::set(), DataHandler::set(), DataHandler::set_join(), DataHandler::set_key(), Form::setFieldAttributes(), FormFieldText::setMaxsize(), FormFieldRadio::setSelected(), FormFieldText::setSize(), setSize(), setValue(), FormFieldRadio::setValue(), Form::showField(), SchemeHandler::table_description(), SchemeHandler::validate_scheme(), and DbHandler::write().

6.14.3.13 BaseElement::setClass (\$ _ value) [inherited]

Set the element Class

Parameters

← \$ _ value Class name

6.14.3.14 FormField::setDisabled (\$ _ value = true) [inherited]

Set the Disabled boolean

Parameters

← \$ _ value Value indicating true (default) or false

6.14.3.15 BaseElement::setId (\$ _ value) [inherited]

Set the element ID

Parameters

← \$ _ value Identification

6.14.3.16 FormFieldSelect::setMultiple (\$ _ value = true)

Set the Multiple boolean

Parameters

← \$ _ value Value indicating true (default) or false

6.14.3.17 FormField::setName (\$ _ value) [inherited]

Set the field name

Parameters

← \$ _ value Field name

6.14.3.18 FormField::set Readonly (\$ _ value = true) [inherited]

Set the Readonly boolean

Parameters

← \$ _ value Value indicating true (default) or false

6.14.3.19 FormFieldSelect::setSize (\$size)

Set the Size attribute

Parameters

$\leftarrow \$size$ integer

References \$size, and `_OWL::set_status()`.

6.14.3.20 FormFieldSelect::setValue (\$value)

Define the options

Parameters

$\leftarrow \$value$ 2-Dimensional array with option data. The first level contains indexed arrays with data for each option in the format:

```
array (
    'value'    => string // Required, value that will be submitted
    , 'text'    => string // Optional, text that will be displayed, defaults to
        'value'
    , 'selected' => boolean // Optional, true when this option is selected, default
        s to false
    , 'class'   => string // Optional, class name
    , 'group'   => string // Optional, label of the optgroup the option belongs
        to
)
```

Reimplemented from [FormField](#).

References `_OWL::set_status()`.

6.14.3.21 _OWL::signal (\$level = OWL_INFO, &\$text = false) [inherited]

Display the message for the current object status

Parameters

$\leftarrow \$level$ An optional severity level; message will only be displayed when it is at least of this level.

$\$text$ If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References `ConfigHandler::get()`.

Referenced by `_OWL::set_status()`, and `_OWL::traceback()`.

6.14.3.22 _OWL::traceback (&\$text = false, \$depth = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References `_OWL::signal()`.

Referenced by `_OWL::check()`, `UserHandler::login()`, and `SessionHandler::read()`.

6.14.4 Member Data Documentation

6.14.4.1 BaseElement::\$class = " [protected, inherited]

Class specification

6.14.4.2 FormField::\$disabled [protected, inherited]

Boolean indicating a disabled field when true

6.14.4.3 BaseElement::\$id = " [protected, inherited]

Element ID

6.14.4.4 FormFieldSelect::\$multiple [private]

Boolean indicating a multiple select

6.14.4.5 FormField::\$name [protected, inherited]

Name of the formfield

6.14.4.6 FormFieldSelect::\$options [private]

Array with options for the select list

6.14.4.7 FormField::\$readonly [protected, inherited]

Boolean indicating a readonly field when true

6.14.4.8 _OWL::\$severity [protected, inherited]

Severity level of the current object status

6.14.4.9 FormFieldSelect::\$size [private]

Number of visible options

Referenced by `setSize()`.

6.14.4.10 BaseElement::\$style = "" [protected, inherited]

Element style

6.14.4.11 FormField::\$type [protected, inherited]

Field type

Reimplemented in [FormFieldTextarea](#).

Referenced by `FormFieldText::__construct()`, and `FormFieldButton::__construct()`.

6.14.4.12 FormField::\$value [protected, inherited]

Field value

6.14.4.13 const FormFieldSelect::DefaultOptionGroup = '___OWL_OptGroup___'

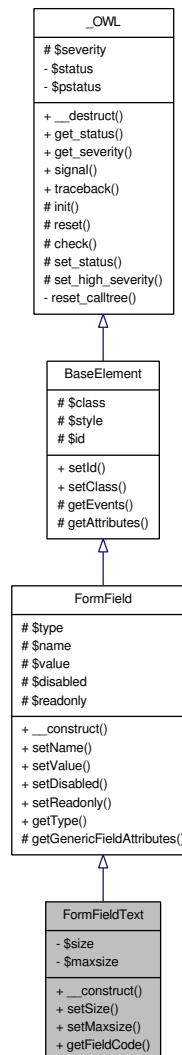
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/ui/formfields/[class.formfield.select.php](#)

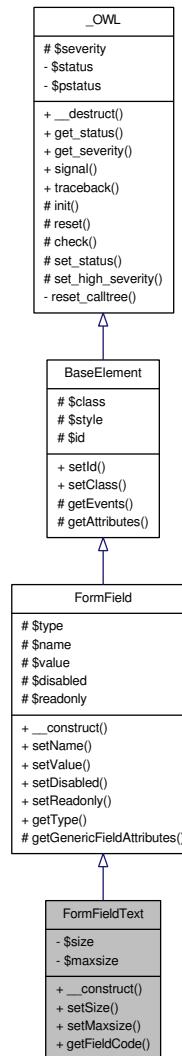
6.15 FormFieldText Class Reference

Formfield.

Inheritance diagram for FormFieldText:



Collaboration diagram for FormFieldText:



Public Member Functions

- **__construct** (\$type= 'text')
- **setSize** (\$size)
- **setMaxsize** (\$maxsize)
- **getFieldCode** ()
- **setName** (\$_value)
- **setValue** (\$_value)
- **setDisabled** (\$_value=true)
- **setReadonly** (\$_value=true)
- **getType** ()
- **setId** (\$_value)
- **setClass** (\$_value)
- **get_status** ()
- **get_severity** (\$status=null)

- `signal` (\$level=OWL_INFO, &\$text=false)
- `traceback` (&\$text=false, \$depth=0)

Protected Member Functions

- `getGenericFieldAttributes` (\$_ignore=array())
- `getEvents` ()
- `getAttributes` ()
- `init` ()
- `reset` ()
- `check` (&\$object, \$level)
- `set_status` (\$status, \$params=array())
- `set_high_severity` (&\$object=null)

Protected Attributes

- `$type`
- `$name`
- `$value`
- `$disabled`
- `$readonly`
- `$class = ''`
- `$style = ''`
- `$id = ''`
- `$severity`

Private Attributes

- `$size`
- `$maxsize`

6.15.1 Detailed Description

Formfield. Formfield text elements

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Oct 19, 2010 -- O van Eijk -- initial version

6.15.2 Constructor & Destructor Documentation

6.15.2.1 FormFieldText::__construct (\$type = 'text')

Class constructor;

Parameters

← **\$type** Element type: text (default), password or hidden

References FormField::\$type, and FormField::__construct().

6.15.3 Member Function Documentation

6.15.3.1 _OWL::check (&\$ object, \$ level) [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

← **\$object** Pointer to an object to check against

← **\$level** The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References _OWL::reset(), _OWL::set_high_severity(), and _OWL::traceback().

Referenced by SessionHandler::write().

6.15.3.2 _OWL::get_severity (\$ status = null) [inherited]

Get the current object severity level.

Parameters

← **\$status** An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References _OWL::\$status.

Referenced by LogHandler::compose_message(), LogHandler::log(), and DataHandler::set_key().

6.15.3.3 _OWL::get_status () [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by SchemeHandler::compare().

6.15.3.4 BaseElement::getAttributes () [protected, inherited]

Return the general element attributes that are set

Returns

string attributes in HTML format (' fld="value"...')

Referenced by Form::openForm().

6.15.3.5 BaseElement::getEvents () [protected, inherited]

Referenced by FormField::getGenericFieldAttributes().

6.15.3.6 FormFieldText::getFieldCode ()

Return the HTML code to display the form element

Returns

Textstring with the complete code for the form element

References FormField::getGenericFieldAttributes().

6.15.3.7 FormField::getGenericFieldAttributes (\$ _ignore = array()) [protected, inherited]

Return the attributes for an HTML formfield in the " attrib='value' [...]" format

Parameters

← *\$ _ignore* Array with attributes names that should be ignored, e.g. for a textarea, the value is not returned as an attribute.

Returns

Textstring with the HTML code

References BaseElement::getEvents().

Referenced by FormFieldTextarea::getFieldCode(), getFieldCode(), FormFieldSelect::getFieldCode(), FormFieldRadio::getFieldCode(), FormFieldCheckbox::getFieldCode(), and FormFieldButton::getFieldCode().

6.15.3.8 FormField::getType () [inherited]

Give the field type

Returns

Field type

6.15.3.9 `_OWL::init ()` [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References `OWL::factory()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `LogHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `OWL::__construct()`, `Dispatcher::__construct()`, and `DataHandler::DataHandler()`.

6.15.3.10 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in `DbHandler`, and `SchemeHandler`.

References `_OWL::reset_calltree()`.

Referenced by `_OWL::check()`, `SessionHandler::read()`, `DataHandler::reset()`, and `_OWL::set_status()`.

6.15.3.11 `_OWL::set_high_severity (&$ object = null)` [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.15.3.12 `_OWL::set_status ($status, $params = array ())` [final, protected, inherited]

Set the current object status to the specified value.

Parameters

$\leftarrow \$status$ `OWL` status code

$\leftarrow \$params$

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler::DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`, `Dispatcher::dispatch()`, `FormHandler::get()`, `DataHandler::get()`, `SchemeHandler::get_table_columns()`, `SchemeHandler::get_table_indexes()`, `UserHandler::login()`, `FileHandler::open()`, `DbHandler::open()`, `LogHandler::open_logfile()`, `DataHandler::prepare()`, `DbHandler::prepare_delete()`, `DbHandler::prepare_insert()`, `DbHandler::prepare_read()`, `DbHandler::prepare_update()`, `DbHandler::read()`, `FileHandler::read_line()`, `UserHandler::read_userdata()`,

DataHandler::reset(), SchemeHandler::scheme(), FormHandler::set(), DataHandler::set(), DataHandler::set_join(), DataHandler::set_key(), Form::setFieldAttributes(), setMaxsize(), FormFieldRadio::setSelected(), setSize(), FormFieldSelect::setSize(), FormFieldSelect::setValue(), FormFieldRadio::setValue(), Form::showField(), SchemeHandler::table_description(), SchemeHandler::validate_scheme(), and DbHandler::write().

6.15.3.13 BaseElement::setClass (\$ _ value) [inherited]

Set the element Class

Parameters

$\leftarrow \$_value$ Class name

6.15.3.14 FormField::setDisabled (\$ _ value = true) [inherited]

Set the Disabled boolean

Parameters

$\leftarrow \$_value$ Value indicating true (default) or false

6.15.3.15 BaseElement::setId (\$ _ value) [inherited]

Set the element ID

Parameters

$\leftarrow \$_value$ Identification

6.15.3.16 FormFieldText::setMaxsize (\$ maxsize)

Set the Maxsize attribute

Parameters

$\leftarrow \$maxsize$ integer

References \$maxsize, and _OWL::set_status().

6.15.3.17 FormField::setName (\$ _ value) [inherited]

Set the field name

Parameters

$\leftarrow \$_value$ Field name

6.15.3.18 FormField::set Readonly (\$ _ value = true) [inherited]

Set the Readonly boolean

Parameters

$\leftarrow \$_value$ Value indicating true (default) or false

6.15.3.19 FormFieldText::setSize (\$ size)

Set the Size attribute

Parameters

$\leftarrow \$size$ integer

References \$size, and `_OWL::set_status()`.

6.15.3.20 FormField::setValue (\$ _ value) [inherited]

Set the field value

Parameters

$\leftarrow \$_value$ Field value

Reimplemented in [FormFieldCheckbox](#), [FormFieldRadio](#), and [FormFieldSelect](#).

6.15.3.21 _OWL::signal (\$ level = OWL_INFO, &\$ text = false) [inherited]

Display the message for the current object status

Parameters

$\leftarrow \$level$ An optional severity level; message will only be displayed when it is at least of this level.

$\$text$ If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by `_OWL::set_status()`, and `_OWL::traceback()`.

6.15.3.22 _OWL::traceback (&\$ text = false, \$ depth = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

- ← **\$text** Optional variable in which the message text can be stored. If not given, the text will be written to standard output
- ← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References `_OWL::signal()`.

Referenced by `_OWL::check()`, `UserHandler::login()`, and `SessionHandler::read()`.

6.15.4 Member Data Documentation

6.15.4.1 BaseElement::\$class = " [protected, inherited]

Class specification

6.15.4.2 FormField::\$disabled [protected, inherited]

Boolean indicating a disabled field when true

6.15.4.3 BaseElement::\$id = " [protected, inherited]

Element ID

6.15.4.4 FormFieldText::\$maxsize [private]

Maximum field size

Referenced by `setMaxsize()`.

6.15.4.5 FormField::\$name [protected, inherited]

Name of the formfield

6.15.4.6 FormField::\$readonly [protected, inherited]

Boolean indicating a readonly field when true

6.15.4.7 _OWL::\$severity [protected, inherited]

Severity level of the current object status

6.15.4.8 FormFieldText::\$size [private]

Field size

Referenced by `setSize()`.

6.15.4.9 BaseElement::\$style = "" [protected, inherited]

Element style

6.15.4.10 FormField::\$type [protected, inherited]

Field type

Reimplemented in [FormFieldTextarea](#).

Referenced by `__construct()`, and `FormFieldButton::__construct()`.

6.15.4.11 FormField::\$value [protected, inherited]

Field value

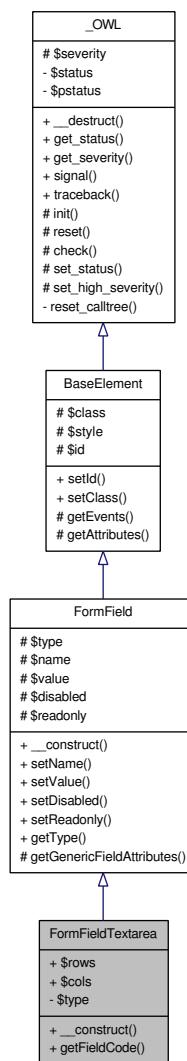
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/ui/formfields/[class.formfield.text.php](#)

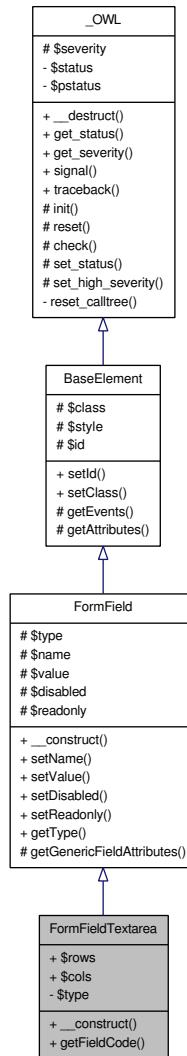
6.16 FormFieldTextarea Class Reference

Formfield.

Inheritance diagram for FormFieldTextarea:



Collaboration diagram for FormFieldTextarea:



Public Member Functions

- `__construct ()`
- `getFieldCode ()`
- `setName ($_value)`
- `setValue ($_value)`
- `setDisabled ($_value=true)`
- `setReadonly ($_value=true)`
- `getType ()`
- `setId ($_value)`
- `setClass ($_value)`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Public Attributes

- `$rows`
- `$cols`

Protected Member Functions

- `getGenericFieldAttributes` (`$_ignore=array()`)
- `getEvents` ()
- `getAttributes` ()
- `init` ()
- `reset` ()
- `check` (&\$object, \$level)
- `set_status` (\$status, \$params=array())
- `set_high_severity` (&\$object=null)

Protected Attributes

- `$name`
- `$value`
- `$disabled`
- `$readonly`
- `$class = ''`
- `$style = ''`
- `$id = ''`
- `$severity`

Private Attributes

- `$type`

6.16.1 Detailed Description

Formfield. Formfield textarea elements

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Oct 19, 2010 -- O van Eijk -- initial version

6.16.2 Constructor & Destructor Documentation

6.16.2.1 FormFieldTextarea::__construct ()

Class constructor;

Reimplemented from [FormField](#).

6.16.3 Member Function Documentation

6.16.3.1 `_OWL::check (&$ object, $ level)` [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← `$object` Pointer to an object to check against
- ← `$level` The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.16.3.2 `_OWL::get_severity ($ status = null)` [inherited]

Get the current object severity level.

Parameters

- ← `$status` An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References `_OWL::$status`.

Referenced by `LogHandler::compose_message()`, `LogHandler::log()`, and `DataHandler::set_key()`.

6.16.3.3 `_OWL::get_status ()` [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by `SchemeHandler::compare()`.

6.16.3.4 `BaseElement::getAttributes ()` [protected, inherited]

Return the general element attributes that are set

Returns

string attributes in HTML format (' fld="value"...')

Referenced by `Form::openForm()`.

6.16.3.5 BaseElement::getEvents () [protected, inherited]

Referenced by FormField::getGenericFieldAttributes().

6.16.3.6 FormFieldTextarea::getFieldCode ()

Return the HTML code to display the form element

Returns

Textstring with the complete code for the form element

References FormField::getGenericFieldAttributes().

6.16.3.7 FormField::getGenericFieldAttributes (\$ _ignore = array()) [protected, inherited]

Return the attributes for an HTML formfield in the " attrib='value' [...]" format

Parameters

← *\$ _ignore* Array with attributes names that should be ignored, e.g. for a textarea, the value is not returned as an attribute.

Returns

Textstring with the HTML code

References BaseElement::getEvents().

Referenced by getFieldCode(), FormFieldText::getFieldCode(), FormFieldSelect::getFieldCode(), FormFieldRadio::getFieldCode(), FormFieldCheckbox::getFieldCode(), and FormFieldButton::getFieldCode().

6.16.3.8 FormField::getType () [inherited]

Give the field type

Returns

Field type

6.16.3.9 _OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References OWL::factory().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), LogHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), OWL::__construct(), Dispatcher::__construct(), and DataHandler::DataHandler().

6.16.3.10 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References `_OWL::reset_calltree()`.

Referenced by `_OWL::check()`, `SessionHandler::read()`, `DataHandler::reset()`, and `_OWL::set_status()`.

6.16.3.11 `_OWL::set_high_severity (&$ object = null)` [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.16.3.12 `_OWL::set_status ($ status, $ params = array ())` [final, protected, inherited]

Set the current object status to the specified value.

Parameters

$\leftarrow \$status$ [OWL](#) status code

$\leftarrow \$params$

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler::DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`, `Dispatcher::dispatch()`, `FormHandler::get()`, `DataHandler::get()`, `SchemeHandler::get_table_columns()`, `SchemeHandler::get_table_indexes()`, `UserHandler::login()`, `FileHandler::open()`, `DbHandler::open()`, `LogHandler::open_logfile()`, `DataHandler::prepare()`, `DbHandler::prepare_delete()`, `DbHandler::prepare_insert()`, `DbHandler::prepare_read()`, `DbHandler::prepare_update()`, `DbHandler::read()`, `FileHandler::read_line()`, `UserHandler::read_userdata()`, `DataHandler::reset()`, `SchemeHandler::scheme()`, `FormHandler::set()`, `DataHandler::set()`, `DataHandler::set_join()`, `DataHandler::set_key()`, `Form::setFieldAttributes()`, `FormFieldText::setMaxsize()`, `FormFieldRadio::setSelected()`, `FormFieldText::setSize()`, `FormFieldSelect::setSize()`, `FormFieldSelect::setValue()`, `FormFieldRadio::setValue()`, `Form::showField()`, `SchemeHandler::table_description()`, `SchemeHandler::validate_scheme()`, and `DbHandler::write()`.

6.16.3.13 `BaseElement::setClass ($ _value)` [inherited]

Set the element Class

Parameters

$\leftarrow \$_value$ Class name

6.16.3.14 FormField::setDisabled (\$ _value = true) [inherited]

Set the Disabled boolean

Parameters

← *\$ _value* Value indicating true (default) or false

6.16.3.15 BaseElement::setId (\$ _value) [inherited]

Set the element ID

Parameters

← *\$ _value* Identification

6.16.3.16 FormField::setName (\$ _value) [inherited]

Set the field name

Parameters

← *\$ _value* Field name

6.16.3.17 FormField::set Readonly (\$ _value = true) [inherited]

Set the Readonly boolean

Parameters

← *\$ _value* Value indicating true (default) or false

6.16.3.18 FormField::setValue (\$ _value) [inherited]

Set the field value

Parameters

← *\$ _value* Field value

Reimplemented in [FormFieldCheckbox](#), [FormFieldRadio](#), and [FormFieldSelect](#).

6.16.3.19 _OWL::signal (\$ level = OWL_INFO, &\$ text = false) [inherited]

Display the message for the current object status

Parameters

← *\$level* An optional severity level; message will only be displayed when it is at least of this level.

\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by _OWL::set_status(), and _OWL::traceback().

6.16.3.20 _OWL::traceback (&\$ *text* = false, \$ *depth* = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References _OWL::signal().

Referenced by _OWL::check(), UserHandler::login(), and SessionHandler::read().

6.16.4 Member Data Documentation**6.16.4.1 BaseElement::\$class = ” [protected, inherited]**

Class specification

6.16.4.2 FormFieldTextarea::\$cols

Number of rows in the textarea

6.16.4.3 FormField::\$disabled [protected, inherited]

Boolean indicating a disabled field when true

6.16.4.4 BaseElement::\$id = ” [protected, inherited]

Element ID

6.16.4.5 FormField::\$name [protected, inherited]

Name of the formfield

6.16.4.6 FormField::\$readonly [protected, inherited]

Boolean indicating a readonly field when true

6.16.4.7 FormFieldTextarea::\$rows

Number of columns in the textarea

6.16.4.8 _OWL::\$severity [protected, inherited]

Severity level of the current object status

6.16.4.9 BaseElement::\$style = "" [protected, inherited]

Element style

6.16.4.10 FormFieldTextarea::\$type [private]

Field type; this class is used for 'text' and 'password' types

Reimplemented from [FormField](#).

6.16.4.11 FormField::\$value [protected, inherited]

Field value

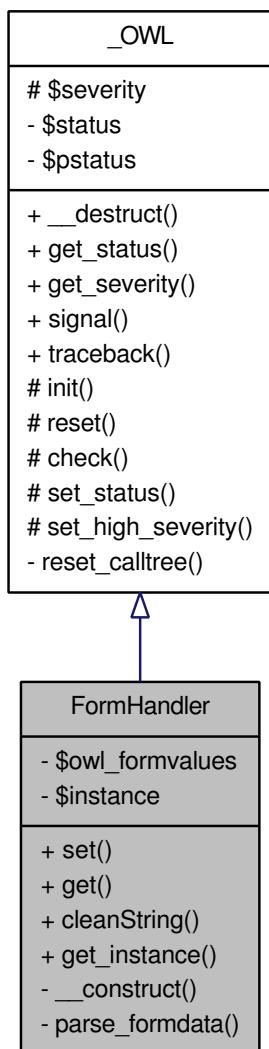
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/ui/formfields/[class.formfield.textarea.php](#)

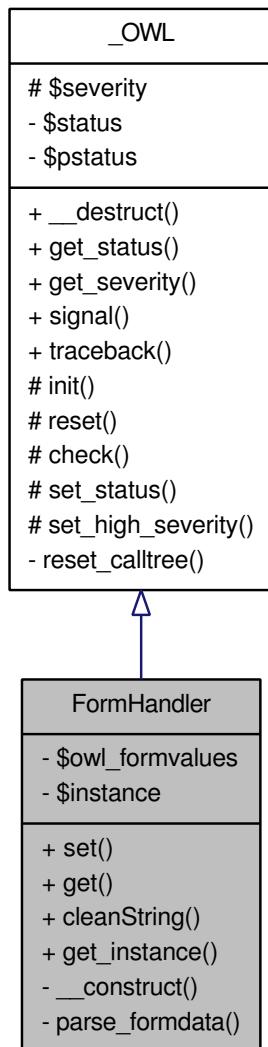
6.17 FormHandler Class Reference

Formhandler.

Inheritance diagram for FormHandler:



Collaboration diagram for FormHandler:



Public Member Functions

- `set ($variable, $value)`
- `get ($variable, $format=FORMDATA_STRICT, $allows=null, $content=array('script'))`
- `cleanString ($input, $allow=null, $content=null, $remove_comment=true)`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Static Public Member Functions

- static `get_instance ()`

Protected Member Functions

- `init ()`
- `reset ()`
- `check (&$object, $level)`
- `set_status ($status, $params=array())`
- `set_high_severity (&$object=null)`

Protected Attributes

- `$severity`

Private Member Functions

- `__construct ()`
- `parse_formdata ($data=null)`

Private Attributes

- `$owl_formvalues`

Static Private Attributes

- static `$instance`

6.17.1 Detailed Description

Formhandler. Handler for all incoming formdata.

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Aug 28, 2008 -- O van Eijk -- initial version

6.17.2 Constructor & Destructor Documentation

6.17.2.1 FormHandler::__construct () [private]

Class constructor; Parse the incoming formdata.

References `_OWL::init()`, `parse_formdata()`, and `_OWL::set_status()`.

6.17.3 Member Function Documentation

6.17.3.1 `_OWL::check (&$ object, $ level)` [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← **\$object** Pointer to an object to check against
- ← **\$level** The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.17.3.2 `FormHandler::cleanString ($ input, $ allow = null, $ content = null, $ remove_comment = true)`

Remove all tags from an input string

Parameters

- ← **\$input** Input string to clean
- ← **\$allow** array with tags that should not be removed. These can be in a regular expression format, e.g. `array('b', 'h[1-6]')` to allow the `b>` tag and `h1>` to `h6>`.
- ← **\$content** array with tags that should completely be removed, so including all contents between the opening and closing tag, e.g. `'script>some code /script>'`. Default is `array('script', 'style')`
- ← **\$remove_comment** When true (default), content blocks will completely be removed. Set to false to allow the content of a comment block. The tags themselves ('`!--`' and '`-->`') will always be removed.

Returns

The string with all tags removed

Referenced by `get()`.

6.17.3.3 `FormHandler::get ($ variable, $ format = FORMDATA_STRICT, $ allows = null, $ content = array('script'))`

Get the value of a given formfield, formatted as specified

Parameters

- ← **\$variable** The variable name who's value should be returned
- ← **\$format** Specify how the field should be formatted

← *\$allows* Array with allowd tags, used for the FORMDATA_CUSTOM format.
 ← *\$content* Array with tags to allow completely, used for the FORMDATA_CUSTOM format.

Returns

Value as taken from the form; single value or array

References cleanString(), ConfigHandler::get(), and _OWL::set_status().

6.17.3.4 static FormHandler::get_instance () [static]

Return a reference to my implementation. If necessary, create that implementation first.

Returns

Severity level

References \$instance.

6.17.3.5 _OWL::get_severity (\$status = null) [inherited]

Get the current object severity level.

Parameters

← *\$status* An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References _OWL::\$status.

Referenced by LogHandler::compose_message(), LogHandler::log(), and DataHandler::set_key().

6.17.3.6 _OWL::get_status () [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by SchemeHandler::compare().

6.17.3.7 _OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the contructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References OWL::factory().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), LogHandler::__construct(), ImageHandler::__construct(), __construct(), FileHandler::__construct(), DbHandler::__construct(), OWL::__construct(), Dispatcher::__construct(), and DataHandler::DataHandler().

6.17.3.8 FormHandler::parse_formdata (\$ data = null) [private]

Parse a given form and store all data in the parent class, except values that come from a multiple select; they will be stored locally.

Parameters

← **\$data** The formdata array.

Referenced by __construct().

6.17.3.9 _OWL::reset () [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References _OWL::reset_calltree().

Referenced by _OWL::check(), SessionHandler::read(), DataHandler::reset(), and _OWL::set_status().

6.17.3.10 FormHandler::set (\$ variable, \$ value)

Reimplement the get method; the parent's get will only be called if the requested variable name is not in the 'local' array where multi-values are stored.

Parameters

← **\$variable** The variable name who's value should be set

← **\$value** Value for the variable

References ConfigHandler::get(), and _OWL::set_status().

6.17.3.11 _OWL::set_high_severity (&\$ object = null) [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by _OWL::check(), DataHandler::db(), DataHandler::prepare(), and SessionHandler::read().

6.17.3.12 _OWL::set_status (\$ status, \$ params = array ()) [final, protected, inherited]

Set the current object status to the specified value.

Parameters

← **\$status** OWL status code
 ← **\$params**

References \$GLOBALS, _OWL::\$status, ConfigHandler::get(), Register::get_code(), _OWL::reset(), and _OWL::signal().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), ImageHandler::__construct(), __construct(), FileHandler::__construct(), DbHandler::__construct(), Form::addField(), SchemeHandler::alter_scheme(), FileHandler::close(), DbHandler::connect(), DbHandler::create(), SchemeHandler::create_scheme(), DataHandler::DataHandler(), SchemeHandler::define_index(), SchemeHandler::define_scheme(), Dispatcher::dispatch(), get(), DataHandler::get(), SchemeHandler::get_table_columns(), SchemeHandler::get_table_indexes(), UserHandler::login(), FileHandler::open(), DbHandler::open(), LogHandler::open_logfile(), DataHandler::prepare(), DbHandler::prepare_delete(), DbHandler::prepare_insert(), DbHandler::prepare_read(), DbHandler::prepare_update(), DbHandler::read(), FileHandler::read_line(), UserHandler::read_userdata(), DataHandler::reset(), SchemeHandler::scheme(), set(), DataHandler::set(), DataHandler::set_join(), DataHandler::set_key(), Form::setFieldAttributes(), FormFieldText::setMaxsize(), FormFieldRadio::setSelected(), FormFieldText::setSize(), FormFieldSelect::setSize(), FormFieldSelect::setValue(), FormFieldRadio::setValue(), Form::showField(), SchemeHandler::table_description(), SchemeHandler::validate_scheme(), and DbHandler::write().

6.17.3.13 _OWL::signal (\$level = OWL_INFO, &\$text = false) [inherited]

Display the message for the current object status

Parameters

← **\$level** An optional severity level; message will only be displayed when it is at least of this level.
\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by _OWL::set_status(), and _OWL::traceback().

6.17.3.14 _OWL::traceback (&\$text = false, \$depth = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output
 ← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References `_OWL::signal()`.

Referenced by `_OWL::check()`, `UserHandler::login()`, and `SessionHandler::read()`.

6.17.4 Member Data Documentation

6.17.4.1 FormHandler::\$instance [static, private]

integer - self reference

Referenced by `get_instance()`.

6.17.4.2 FormHandler::\$owl_formvalues [private]

Create an array which will hold all (formatted) formvalues

6.17.4.3 _OWL::\$severity [protected, inherited]

Severity level of the current object status

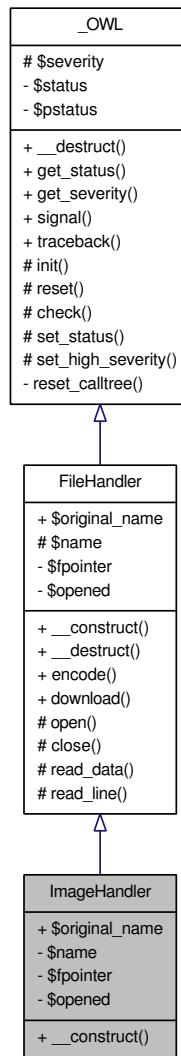
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/so/[class.formhandler.php](#)

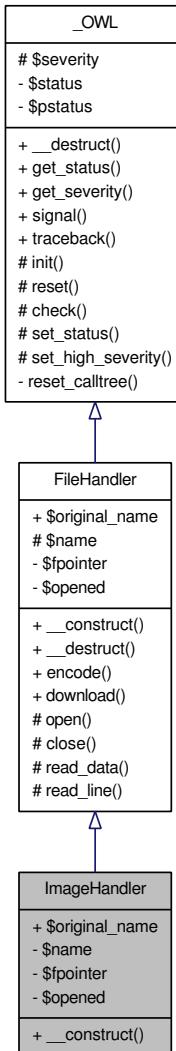
6.18 ImageHandler Class Reference

Image handler.

Inheritance diagram for ImageHandler:



Collaboration diagram for ImageHandler:



Public Member Functions

- `__construct` (\$name, \$req=false)
- `encode` ()
- `download` ()
- `get_status` ()
- `get_severity` (\$status=null)
- `signal` (\$level=OWL_INFO, &\$text=false)
- `traceback` (&\$text=false, \$depth=0)

Public Attributes

- `$original_name`

Protected Member Functions

- `open ($mode= 'r')`
- `close ()`
- `read_data ()`
- `read_line ($trim=FILE_NOTRIM)`
- `init ()`
- `reset ()`
- `check (&$object, $level)`
- `set_status ($status, $params=array())`
- `set_high_severity (&$object=null)`

Protected Attributes

- `$severity`

Private Attributes

- `$name`
- `$fpointer`
- `$opened`

6.18.1 Detailed Description

Image handler. Handle all images

Author

Oscar van Eijk, Oveas Functionality Provider

Version

May 15, 2007 -- O van Eijk -- initial version for Terra-Terra (based on an old OFM module)
 Jul 30, 2008 -- O van Eijk -- Modified version for OWL-PHP

6.18.2 Constructor & Destructor Documentation

6.18.2.1 ImageHandler::__construct (\$ name, \$ req = false)

Class constructor; setup the image characteristics

Parameters

- ← `$name` Filename
- ← `$req` True is the file must exist at object create time

Reimplemented from [FileHandler](#).

References `$name`, `_OWL::init()`, and `_OWL::set_status()`.

6.18.3 Member Function Documentation

6.18.3.1 `_OWL::check (&$ object, $ level)` [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

← `$object` Pointer to an object to check against

← `$level` The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.18.3.2 `FileHandler::close ()` [protected, inherited]

Close the file

References `_OWL::set_status()`.

Referenced by `FileHandler::__destruct()`, and `FileHandler::read_data()`.

6.18.3.3 `FileHandler::download ()` [inherited]

6.18.3.4 `FileHandler::encode ()` [inherited]

References `FileHandler::read_data()`.

6.18.3.5 `_OWL::get_severity ($ status = null)` [inherited]

Get the current object severity level.

Parameters

← `$status` An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References `_OWL::$status`.

Referenced by `LogHandler::compose_message()`, `LogHandler::log()`, and `DataHandler::set_key()`.

6.18.3.6 `_OWL::get_status () [final, inherited]`

Get the current object status.

Returns

Object's status code

Referenced by SchemeHandler::compare().

6.18.3.7 `_OWL::init () [protected, inherited]`

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References OWL::factory().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), LogHandler::__construct(), __construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), OWL::__construct(), Dispatcher::__construct(), and DataHandler::DataHandler().

6.18.3.8 `FileHandler::open ($ mode = 'r') [protected, inherited]`

Open the file

Parameters

← `$mode` Mode in which the file should be opened:

- 'r' Open for reading only; place the file pointer at the beginning of the file.
- 'r+' Open for reading and writing; place the file pointer at the beginning of the file.
- 'w' Open for writing only; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.
- 'w+' Open for reading and writing; place the file pointer at the beginning of the file and truncate the file to zero length. If the file does not exist, attempt to create it.
- 'a' Open for writing only; place the file pointer at the end of the file. If the file does not exist, attempt to create it.
- 'a+' Open for reading and writing; place the file pointer at the end of the file. If the file does not exist, attempt to create it.

References _OWL::set_status().

Referenced by FileHandler::read_data().

6.18.3.9 `FileHandler::read_data () [protected, inherited]`

Read the file contents and return as one dataset

References FileHandler::close(), and FileHandler::open().

Referenced by FileHandler::encode().

6.18.3.10 FileHandler::read_line (\$ trim = FILE_NOTRIM) [protected, inherited]

Read a single line from the file. File must be opened before

Parameters

← *\$trim* specify how the returned line should be trimmed:

- FILE_NOTRIM
- FILE_TRIM_L
- FILE_TRIM_R
- FILE_TRIM_C

References `_OWL::set_status()`.

6.18.3.11 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References `_OWL::reset_calltree()`.

Referenced by `_OWL::check()`, `SessionHandler::read()`, `DataHandler::reset()`, and `_OWL::set_status()`.

6.18.3.12 `_OWL::set_high_severity (&$ object = null)` [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.18.3.13 `_OWL::set_status ($ status, $ params = array ())` [final, protected, inherited]

Set the current object status to the specified value.

Parameters

← *\$status* OWL status code
 ← *\$params*

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `__construct()`, `__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler::DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`, `Dispatcher::dispatch()`, `FormHandler::get()`, `DataHandler::get()`, `SchemeHandler::get_table_columns()`, `SchemeHandler::get_table_indexes()`, `UserHandler::login()`, `FileHandler::open()`,

DbHandler::open(), LogHandler::open_logfile(), DataHandler::prepare(), DbHandler::prepare_delete(), DbHandler::prepare_insert(), DbHandler::prepare_read(), DbHandler::prepare_update(), DbHandler::read(), FileHandler::read_line(), UserHandler::read_userdata(), DataHandler::reset(), SchemeHandler::scheme(), FormHandler::set(), DataHandler::set(), DataHandler::set_join(), DataHandler::set_key(), Form::setFieldAttributes(), FormFieldText::setMaxsize(), FormFieldRadio::setSelected(), FormFieldText::setSize(), FormFieldSelect::setSize(), FormFieldSelect::setValue(), FormFieldRadio::setValue(), Form::showField(), SchemeHandler::table_description(), SchemeHandler::validate_scheme(), and DbHandler::write().

6.18.3.14 `_OWL::signal ($level = OWL_INFO, &$text = false)` [inherited]

Display the message for the current object status

Parameters

← **\$level** An optional severity level; message will only be displayed when it is at least of this level.

\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by _OWL::set_status(), and _OWL::traceback().

6.18.3.15 `_OWL::traceback (&$text = false, $depth = 0)` [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References _OWL::signal().

Referenced by _OWL::check(), UserHandler::login(), and SessionHandler::read().

6.18.4 Member Data Documentation

6.18.4.1 `ImageHandler::$fpointer` [private]

Pointer to the file when opened

Reimplemented from [FileHandler](#).

6.18.4.2 ImageHandler::\$name [private]

Full filename as stored on the file system

Reimplemented from [FileHandler](#).

Referenced by `__construct()`.

6.18.4.3 ImageHandler::\$opened [private]

Boolean that's true when the file is opened

Reimplemented from [FileHandler](#).

6.18.4.4 ImageHandler::\$original_name

Reimplemented from [FileHandler](#).

6.18.4.5 _OWL::\$severity [protected, inherited]

Severity level of the current object status

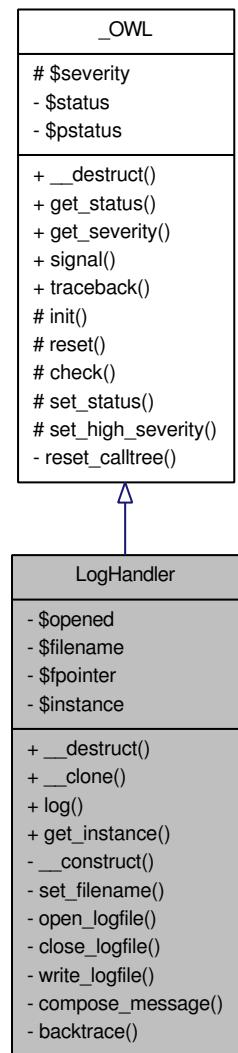
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/so/[class.imagehandler.php](#)

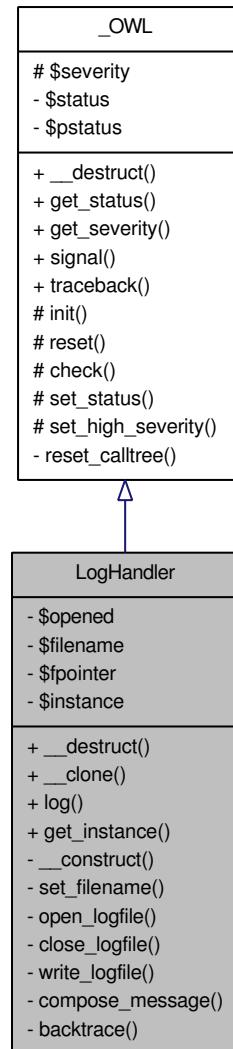
6.19 LogHandler Class Reference

Log handler.

Inheritance diagram for LogHandler:



Collaboration diagram for LogHandler:



Public Member Functions

- `__destruct ()`
- `__clone ()`
- `log ($msg, $code)`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Static Public Member Functions

- static `get_instance ()`

Protected Member Functions

- `init ()`
- `reset ()`
- `check (&$object, $level)`
- `set_status ($status, $params=array())`
- `set_high_severity (&$object=null)`

Protected Attributes

- `$severity`

Private Member Functions

- `__construct ()`
- `set_filename ()`
- `open_logfile ()`
- `close_logfile ()`
- `write_logfile ($msg)`
- `compose_message (&$msg, $code)`
- `backtrace ()`

Private Attributes

- `$opened`
- `$filename`
- `$fpointer`

Static Private Attributes

- static `$instance`

6.19.1 Detailed Description

Log handler. This singleton class handles all [OWL](#) logging

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Aug 13, 2008 -- O van Eijk -- initial version

6.19.2 Constructor & Destructor Documentation

6.19.2.1 LogHandler::__construct () [private]

Class constructor

References `ConfigHandler::get()`, `_OWL::init()`, `open_logfile()`, and `set_filename()`.

6.19.2.2 LogHandler::__destruct ()

Class destructor

Reimplemented from [_OWL](#).

References `close_logfile()`.

6.19.3 Member Function Documentation

6.19.3.1 LogHandler::__clone ()

Implementation of the [__clone\(\)](#) function to prevent cloning of this singleton; it triggers a fatal (user)error

6.19.3.2 LogHandler::backtrace () [private]

Create a backtrace of the current log item

Returns

Trace information of this call.

Referenced by `log()`.

6.19.3.3 _OWL::check (&\$ object, \$ level) [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

← *\$object* Pointer to an object to check against

← *\$level* The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.19.3.4 LogHandler::close_logfile () [private]

Close the logfile

Referenced by `__destruct()`, and `log()`.

6.19.3.5 LogHandler::compose_message (&\$ msg, \$ code) [private]

Compose the logmessage by adding a timestamp and - when not writing multiple files - the run ID

Parameters

\$msg Original message by the event

← *\$code* Status code of the message

References ConfigHandler::get(), Register::get_code(), Register::get_run_id(), and _OWL::get_severity().

Referenced by log().

6.19.3.6 static LogHandler::get_instance () [static]

Return a reference to my implementation. If necessary, create that implementation first.

Returns

Severity level

References \$instance.

6.19.3.7 _OWL::get_severity (\$ status = null) [inherited]

Get the current object severity level.

Parameters

← *\$status* An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References _OWL::\$status.

Referenced by compose_message(), log(), and DataHandler::set_key().

6.19.3.8 _OWL::get_status () [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by SchemeHandler::compare().

6.19.3.9 `_OWL::init ()` [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References `OWL::factory()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `OWL::__construct()`, `Dispatcher::__construct()`, and `DataHandler::DataHandler()`.

6.19.3.10 `LogHandler::log ($ msg, $ code)`

Log an event signalled by [OWL](#)

Parameters

$\leftarrow \$msg$ Message text

$\leftarrow \$code$ Status code of the message

References `backtrace()`, `close_logfile()`, `compose_message()`, `ConfigHandler::get()`, `_OWL::get_severity()`, `open_logfile()`, and `write_logfile()`.

6.19.3.11 `LogHandler::open_logfile ()` [private]

Open the logfile for write

References `_OWL::set_status()`.

Referenced by `__construct()`, and `log()`.

6.19.3.12 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References `_OWL::reset_calltree()`.

Referenced by `_OWL::check()`, `SessionHandler::read()`, `DataHandler::reset()`, and `_OWL::set_status()`.

6.19.3.13 `LogHandler::set_filename ()` [private]

Find out what the filename of the logfile should be

References `ConfigHandler::get()`, and `Register::get_run_id()`.

Referenced by `__construct()`.

6.19.3.14 `_OWL::set_high_severity (&$ object = null)` [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.19.3.15 `_OWL::set_status ($status, $params = array ())` [final, protected, inherited]

Set the current object status to the specified value.

Parameters

- ← `$status` OWL status code
- ← `$params`

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler::DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`, `Dispatcher::dispatch()`, `FormHandler::get()`, `DataHandler::get()`, `SchemeHandler::get_table_columns()`, `SchemeHandler::get_table_indexes()`, `UserHandler::login()`, `FileHandler::open()`, `DbHandler::open()`, `open_logfile()`, `DataHandler::prepare()`, `DbHandler::prepare_delete()`, `DbHandler::prepare_insert()`, `DbHandler::prepare_read()`, `DbHandler::prepare_update()`, `DbHandler::read()`, `FileHandler::read_line()`, `UserHandler::read_userdata()`, `DataHandler::reset()`, `SchemeHandler::scheme()`, `FormHandler::set()`, `DataHandler::set()`, `DataHandler::set_join()`, `DataHandler::set_key()`, `Form::setFieldAttributes()`, `FormFieldText::setMaxsize()`, `FormFieldRadio:: setSelected()`, `FormFieldText::setSize()`, `FormFieldSelect::setSize()`, `FormFieldSelect::setValue()`, `FormFieldRadio::setValue()`, `Form::showField()`, `SchemeHandler::table_description()`, `SchemeHandler::validate_scheme()`, and `DbHandler::write()`.

6.19.3.16 `_OWL::signal ($level = OWL_INFO, &$text = false)` [inherited]

Display the message for the current object status

Parameters

- ← `$level` An optional severity level; message will only be displayed when it is at least of this level.
- `$text` If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References `ConfigHandler::get()`.

Referenced by `_OWL::set_status()`, and `_OWL::traceback()`.

6.19.3.17 `_OWL::traceback (&$text = false, $depth = 0)` [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← *\$depth* This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References `_OWL::signal()`.

Referenced by `_OWL::check()`, `UserHandler::login()`, and `SessionHandler::read()`.

6.19.3.18 LogHandler::write_logfile (\$ msg) [private]

Write a message to the logfile

Parameters

← *\$msg* The complete log message

Referenced by `log()`.

6.19.4 Member Data Documentation**6.19.4.1 LogHandler::\$filename [private]**

Name of the logfile

6.19.4.2 LogHandler::\$fpointer [private]

File pointer

6.19.4.3 LogHandler::\$instance [static, private]

integer - self reference

Referenced by `get_instance()`.

6.19.4.4 LogHandler::\$opened [private]

Boolean to keep track of the logfile status

6.19.4.5 _OWL::\$severity [protected, inherited]

Severity level of the current object status

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

- /home/oscar/projects/owl-php/src/kernel/so/[class.loghandler.php](#)

6.20 OldOFMStuff Class Reference

Public Member Functions

- [OFM_FileDetails](#) (\$Path, \$File)
- [OFM_FileInfo](#) (\$Path, \$File)
- [OFM_IsAscii](#) (\$File, \$Type)

6.20.1 Member Function Documentation

6.20.1.1 OldOFMStuff::OFM_FileDetails (\$ Path, \$ File)

References OFM_FileInfo().

6.20.1.2 OldOFMStuff::OFM_FileInfo (\$ Path, \$ File)

References OFM_IsAscii().

Referenced by OFM_FileDetails().

6.20.1.3 OldOFMStuff::OFM_IsAscii (\$ File, \$ Type)

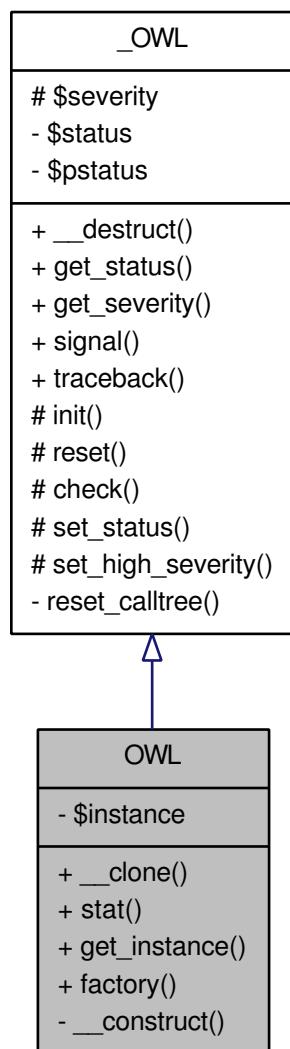
Referenced by OFM_FileInfo().

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

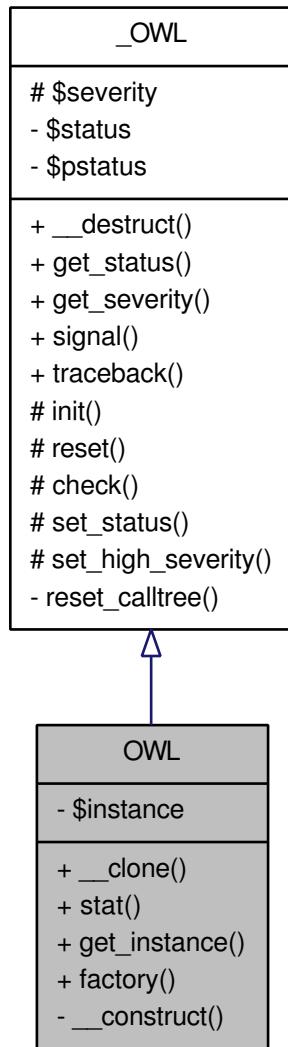
- /home/oscar/projects/owl-php/src/kernel/so/[class.filehandler.php](#)

6.21 OWL Class Reference

Inheritance diagram for OWL:



Collaboration diagram for OWL:



Public Member Functions

- `__clone ()`
- `stat ($a, $b=array())`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Static Public Member Functions

- static `get_instance ()`
- static `factory ($class, $layer= 'so')`

Protected Member Functions

- `init ()`
- `reset ()`
- `check (&$object, $level)`
- `set_status ($status, $params=array())`
- `set_high_severity (&$object=null)`

Protected Attributes

- `$severity`

Private Member Functions

- `__construct ()`

Static Private Attributes

- static `$instance`

6.21.1 Detailed Description

This is a helper class that allows abstract classes to set a status, and provides some standard methods

Author

Oscar van Eijk, Oveas Functionality Provider

6.21.2 Constructor & Destructor Documentation

6.21.2.1 OWL::__construct () [private]

Constructor

References `_OWL::init()`.

6.21.3 Member Function Documentation

6.21.3.1 OWL::__clone ()

Implementation of the `__clone()` function to prevent cloning of this singleton; it triggers a fatal (user)error

6.21.3.2 _OWL::check (&\$ object, \$ level) [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← *\$object* Pointer to an object to check against
- ← *\$level* The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.21.3.3 static OWL::factory (\$ class, \$ layer = 'so') [static]

Loader function to instantiate the singletons or get the existing reference.

Parameters

- ← *\$class* Classname
- ← *\$layer* Layer, defaults to 'so'

Returns

Reference to the object

Referenced by `SessionHandler::__construct()`, `DataHandler::DataHandler()`, `Dispatcher::dispatch()`, and `_OWL::init()`.

6.21.3.4 static OWL::get_instance () [static]

Return a reference to my implementation. If necessary, create that implementation first.

Returns

Severity level

References `$instance`.

Referenced by `stat()`.

6.21.3.5 _OWL::get_severity (\$ status = null) [inherited]

Get the current object severity level.

Parameters

- ← *\$status* An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References `_OWL::$status`.

Referenced by `LogHandler::compose_message()`, `LogHandler::log()`, and `DataHandler::set_key()`.

6.21.3.6 `_OWL::get_status ()` [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by [SchemeHandler::compare\(\)](#).

6.21.3.7 `_OWL::init ()` [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References [factory\(\)](#).

Referenced by [UserHandler::__construct\(\)](#), [SessionHandler::__construct\(\)](#), [SchemeHandler::__construct\(\)](#), [LogHandler::__construct\(\)](#), [ImageHandler::__construct\(\)](#), [FormHandler::__construct\(\)](#), [FileHandler::__construct\(\)](#), [DbHandler::__construct\(\)](#), [__construct\(\)](#), [Dispatcher::__construct\(\)](#), and [DataHandler::DataHandler\(\)](#).

6.21.3.8 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References [_OWL::reset_calltree\(\)](#).

Referenced by [_OWL::check\(\)](#), [SessionHandler::read\(\)](#), [DataHandler::reset\(\)](#), and [_OWL::set_status\(\)](#).

6.21.3.9 `_OWL::set_high_severity (&$ object = null) [protected, inherited]`

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by [_OWL::check\(\)](#), [DataHandler::db\(\)](#), [DataHandler::prepare\(\)](#), and [SessionHandler::read\(\)](#).

6.21.3.10 `_OWL::set_status ($status, $params = array ()) [final, protected, inherited]`

Set the current object status to the specified value.

Parameters

← `$status` [OWL](#) status code

← `$params`

References [\\$GLOBALS](#), [_OWL::\\$status](#), [ConfigHandler::get\(\)](#), [Register::get_code\(\)](#), [_OWL::reset\(\)](#), and [_OWL::signal\(\)](#).

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), Form::addField(), SchemeHandler::alter_scheme(), FileHandler::close(), DbHandler::connect(), DbHandler::create(), SchemeHandler::create_scheme(), DataHandler::DataHandler(), SchemeHandler::define_index(), SchemeHandler::define_scheme(), Dispatcher::dispatch(), FormHandler::get(), DataHandler::get(), SchemeHandler::get_table_columns(), SchemeHandler::get_table_indexes(), UserHandler::login(), FileHandler::open(), DbHandler::open(), LogHandler::open_logfile(), DataHandler::prepare(), DbHandler::prepare_delete(), DbHandler::prepare_insert(), DbHandler::prepare_read(), DbHandler::prepare_update(), DbHandler::read(), FileHandler::read_line(), UserHandler::read_userdata(), DataHandler::reset(), SchemeHandler::scheme(), FormHandler::set(), DataHandler::set(), DataHandler::set_join(), DataHandler::set_key(), Form::setFieldAttributes(), FormFieldText::setMaxsize(), FormFieldRadio::setSelected(), FormFieldText::setSize(), FormFieldSelect::setSize(), FormFieldSelect::setValue(), FormFieldRadio::setValue(), Form::showField(), SchemeHandler::table_description(), SchemeHandler::validate_scheme(), and DbHandler::write().

6.21.3.11 __OWL::signal (\$level = OWL_INFO, &\$text = false) [inherited]

Display the message for the current object status

Parameters

← **\$level** An optional severity level; message will only be displayed when it is at least of this level.

\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by __OWL::set_status(), and __OWL::traceback().

6.21.3.12 OWL::stat (\$a, \$b = array())

Call to [set_status\(\)](#)

Parameters

← **\$a** First parameter for passthrough

← **\$b** Second parameter for passthrough

References get_instance().

Referenced by ConfigHandler::get().

6.21.3.13 __OWL::traceback (&\$text = false, \$depth = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References `_OWL::signal()`.

Referenced by `_OWL::check()`, `UserHandler::login()`, and `SessionHandler::read()`.

6.21.4 Member Data Documentation

6.21.4.1 `OWL::$instance` [static, private]

integer - self reference

Referenced by `get_instance()`.

6.21.4.2 `_OWL::$severity` [protected, inherited]

Severity level of the current object status

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

- /home/oscar/projects/owl-php/src/kernel/bo/[class.owl.php](#)

6.22 OWLException Class Reference

Exception handler.

Public Member Functions

- `__construct` (\$msg=null, \$code=0, `OWLException` \$caller=null)
- `get_trace` ()
- `stack_dump` (\$textmode=true)

Public Attributes

- `$thrown_code`

Private Member Functions

- `check_hide` (\$trace)
- `traceback` (\$trace, \$step, \$textmode)

Private Attributes

- `$caller`
- `$hidden_args`

6.22.1 Detailed Description

Exception handler. Extend the default PHP Exception handler .

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Jul 29, 2008 -- O van Eijk -- Initial version

6.22.2 Constructor & Destructor Documentation

6.22.2.1 `OWLException::__construct` (\$*msg* = null, \$*code* = 0, `OWLException` \$*caller* = null)

Create the Exception handler object

Parameters

- ← `$msg` Message text
- ← `$code` Code of the event
- ← `$caller` Backlink to the calling object

References `$caller`, and `ConfigHandler::get()`.

6.22.3 Member Function Documentation

6.22.3.1 OWLException::check_hide (\$trace) [private]

Check if this traced call has arguments that should be hidden in the stackdump.

Parameters

← **\$trace** The call

Returns

integer holding the argument index that should be hidden, of -1 if nothing has to be hidden.

References ConfigHandler::get().

Referenced by traceback().

6.22.3.2 OWLException::get_trace ()

Trace back the previous object in the stack

Referenced by stack_dump().

6.22.3.3 OWLException::stack_dump (\$textmode = true)

Create an overview of the calling stack

Parameters

← **\$textmode** specify the format in which the stackdump should be returned; text (true, default) or HTML (false).

References Register::get_code(), Register::get_severity(), get_trace(), and traceback().

Referenced by OWLExceptionHandler::log_exception().

6.22.3.4 OWLException::traceback (\$trace, \$step, \$textmode) [private]

Trace a single call from the stack

Parameters

← **\$trace** The call

← **\$step** Number of the call

← **\$textmode** True if the stackdump is created as ASCII text, False for HTML

References check_hide(), and ConfigHandler::get().

Referenced by stack_dump().

6.22.4 Member Data Documentation

6.22.4.1 OWLException::\$caller [private]

Backlink to the calling object

Referenced by `__construct()`.

6.22.4.2 OWLException::\$hidden_args [private]

Array with function call info of which arguments should be hidden

6.22.4.3 OWLException::\$thrown_code

Store the error code to allow the logger to retrieve it

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

- /home/oscar/projects/owl-php/src/kernel/so/[class.exceptionhandler.php](#)

6.23 OWLExceptionHandler Class Reference

Default exception handler.

Static Public Member Functions

- static [log_exception \(OWLException \\$exception\)](#)
- static [handle_exception \(OWLException \\$exception\)](#)

6.23.1 Detailed Description

Default exception handler. Establish a default exception handler. An exception is thrown whenever a status is set above a specified severity level. These exceptions are caught only by this default handler

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Jul 29, 2008 -- O van Eijk -- Initial version

6.23.2 Member Function Documentation

6.23.2.1 static OWLExceptionHandler::handle_exception (OWLException \$exception) [static]

Catch an uncaught exception

Parameters

← *\$exception* The exception

References [log_exception\(\)](#).

6.23.2.2 static OWLExceptionHandler::log_exception (OWLException \$exception) [static]

Show the stackdump of an exception

Parameters

← *\$exception* The exception

References [\\$GLOBALS](#), [ConfigHandler::get\(\)](#), and [OWLException::stack_dump\(\)](#).

Referenced by [handle_exception\(\)](#).

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

- /home/oscar/projects/owl-php/src/kernel/so/[class.exceptionhandler.php](#)

6.24 OWLloader Class Reference

Class loader.

Static Public Member Functions

- static `getClass` (`$_className`, `$_classLocation=array(OWL_SO_INC, OWL_BO_INC, OWL_UI_INC)`, `$_loadMultiple=false`)

Static Private Member Functions

- static `_tryLoad` (`$_className`, `$_classLocation`)

6.24.1 Detailed Description

Class loader. Abstract class to load other classfiles

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Oct 7, 2010 -- O van Eijk -- Initial version for OWL-PHP

6.24.2 Member Function Documentation

6.24.2.1 static OWLloader::_tryLoad (`$_className`, `$_classLocation`) [static, private]

Try loading a class from a specific location

Parameters

- ← `$_className` Name of the classfile
- ← `$_classLocation` Location where to try loading it from

Returns

True on success

References \$GLOBALS.

Referenced by `getClass()`.

6.24.2.2 static OWLloader::getClass (`$_className`, `$_classLocation = array(OWL_SO_INC, OWL_BO_INC, OWL_UI_INC)`, `$_loadMultiple = false`) [static]

Load a PHP classfile

Parameters

- ← ***\$_className*** Name of the class; either file full name (<filename.php> or the identifying name ([class.] name>[.php]))
- ← ***\$_classLocation*** (Array of) location(s) where to look for the class
- ← ***\$_loadMultiple*** Boolean; by default, the first matching classname will be loaded. Set this to true to load all files with the same name from multiple locations

Returns

True on success

References `_tryLoad()`.

Referenced by `Form::addField()`, and `Dispatcher::dispatch()`.

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

- /home/oscar/projects/owl-php/src/[OWLloader.php](#)

6.25 Register Class Reference

Static Public Member Functions

- static `init ()`
- static `register_app ($name, $id)`
- static `register_class ($name)`
- static `register_code ($code)`
- static `register_severity ($level, $name)`
- static `get_severity ($level)`
- static `get_severity_level ($name)`
- static `get_run_id ()`
- static `get_code ($value, $unknown= '*unknown *)'`
- static `set_application ($app_id)`
- static `set_class ($class_id)`
- static `set_severity ($severity_level)`
- static `register_messages ()`

6.25.1 Detailed Description

OWL keeps track of all running applications, their class and all status codes their instances (objects) can have. This is done in a global `Register`, which is maintained by this class.

6.25.2 Member Function Documentation

6.25.2.1 static Register::get_code (\$ value, \$ unknown = '*unknown*)' [static]

Translate an hex value code to the symbolic name

Parameters

- ← `$value` Hex value of the status code
- ← `$unknown` Return value if the code does not exist

Returns

Human readable value

References \$GLOBALS.

Referenced by LogHandler::compose_message(), StatusHandler::get_message(), _OWL::set_status(), and OWLException::stack_dump().

6.25.2.2 static Register::get_run_id () [static]

Return the ID of the current run

References \$GLOBALS.

Referenced by LogHandler::compose_message(), and LogHandler::set_filename().

6.25.2.3 static Register::get_severity (\$ level) [static]

Read a severity level from the register

Parameters

← *\$level* Hex value of the severity level

Returns

Human readable value

References \$GLOBALS.

Referenced by OWLException::stack_dump().

6.25.2.4 static Register::get_severity_level (\$ name) [static]

This function is used by a config parse to translate a string value to the appropriate severity level

Parameters

← *\$name* The name of the severity level

Returns

Hex value of the severity level

References \$GLOBALS.

Referenced by ConfigHandler::convert().

6.25.2.5 static Register::init () [static]

Initialise the register array

References \$GLOBALS.

6.25.2.6 static Register::register_app (\$ name, \$ id) [static]

Store the specified application in the register

Parameters

← *\$name* Name of the class

← *\$id* Application ID

References \$GLOBALS, and set_application().

6.25.2.7 static Register::register_class (\$ name) [static]

Store the specified class in the register, and setup an array to keep track of the codes

Parameters

← *\$name* Name of the class

References \$GLOBALS.

6.25.2.8 static Register::register_code (\$ code) [static]

Define a new statuscode

Parameters

← *\$code* Symbolic name of the status code

References \$GLOBALS.

6.25.2.9 static Register::register_messages () [static]

Load the message file

References ConfigHandler::get().

Referenced by StatusHandler::get_message().

6.25.2.10 static Register::register_severity (\$ level, \$ name) [static]

Store the known severitylevels in the register

Parameters

← *\$level* Symbolic name for the severity level

← *\$name* Human readable value

References \$GLOBALS.

6.25.2.11 static Register::set_application (\$ app_id) [static]

Point the register to the specified application.

Parameters

← *\$app_id* Application ID

References \$GLOBALS.

Referenced by register_app().

6.25.2.12 static Register::set_class (\$ class_id) [static]

Point the register to the specified class.

Parameters

← *\$class_id* Class ID

References \$GLOBALS.

6.25.2.13 static Register::set_severity (\$severity_level) [static]

Set the current severity to the specified level in the [Register](#)

Parameters

← *\$severity_level* Severity level

References \$GLOBALS.

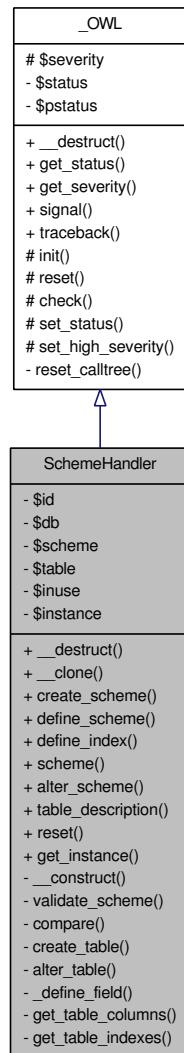
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/so/[class.register.php](#)

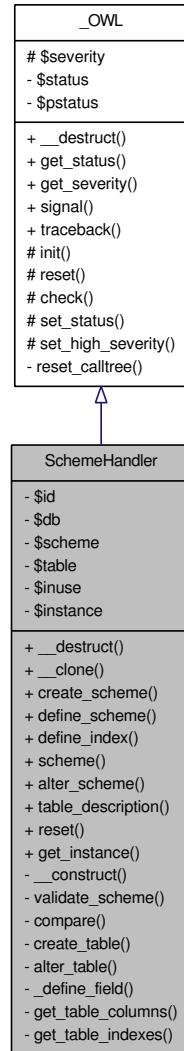
6.26 SchemeHandler Class Reference

Scheme handler.

Inheritance diagram for SchemeHandler:



Collaboration diagram for SchemeHandler:



Public Member Functions

- `__destruct ()`
- `__clone ()`
- `create_scheme ($_tblname)`
- `define_scheme ($_scheme)`
- `define_index ($_index)`
- `scheme ($_drops=false)`
- `alter_scheme ($_field)`
- `table_description ($tablename, &$data)`
- `reset ()`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Static Public Member Functions

- static `get_instance ()`

Protected Member Functions

- `init ()`
- `check (&$object, $level)`
- `set_status ($status, $params=array())`
- `set_high_severity (&$object=null)`

Protected Attributes

- `$severity`

Private Member Functions

- `__construct ()`
- `validate_scheme ()`
- `compare ()`
- `create_table ()`
- `alter_table ($_diffs, $_drops)`
- `_define_field ($_desc)`
- `get_table_columns ($_tablename)`
- `get_table_indexes ($_tablename)`

Private Attributes

- `$id`
- `$db`
- `$scheme`
- `$table = ''`
- `$inuse = false`

Static Private Attributes

- static `$instance`

6.26.1 Detailed Description

Scheme handler. Handler for all database schemes. This singleton class handles all updates to db tables

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Oct 7, 2010 -- O van Eijk -- initial version for [OWL](#)

6.26.2 Constructor & Destructor Documentation

6.26.2.1 SchemeHandler::__construct () [private]

Class constructor

References get_instance(), _OWL::init(), and _OWL::set_status().

6.26.2.2 SchemeHandler::__destruct ()

Class destructor

Reimplemented from [_OWL](#).

6.26.3 Member Function Documentation

6.26.3.1 SchemeHandler::__clone ()

Implementation of the [__clone\(\)](#) function to prevent cloning of this singleton; it triggers a fatal (user)error

6.26.3.2 SchemeHandler::_define_field (\$ _desc) [private]

Create the SQL code for a field definition

Parameters

← *\$ _desc* Indexed array with the field properties from scheme definition

Returns

string SQL code

Referenced by alter_table(), and create_table().

6.26.3.3 SchemeHandler::alter_scheme (\$ _field)

Modify 1 or more fields in an existing scheme definition

Parameters

← *\$ _field* Array holding 1 or more field descriptions

See also

[define_scheme\(\)](#)

Returns

Boolean; false if the table description contains errors

References scheme(), _OWL::set_status(), and validate_scheme().

6.26.3.4 SchemeHandler::alter_table (\$ _diffs, \$ _drops) [private]

Make changes to the table

Parameters

- ← *\$ _diffs* Changes to make
- ← *\$ _drops* True if existing fields should be dropped

References `_define_field()`.

Referenced by `scheme()`.

6.26.3.5 _OWL::check (&\$ object, \$ level) [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← *\$object* Pointer to an object to check against
- ← *\$level* The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.26.3.6 SchemeHandler::compare () [private]

Compare the scheme with an existing database table

Returns

`mixed` True if there are no differences, False if the table does not exist or an array with differences

References `_OWL::get_status()`, `scheme()`, and `table_description()`.

Referenced by `scheme()`.

6.26.3.7 SchemeHandler::create_scheme (\$ _tblname)

Set a new tablename

Parameters

- ← *\$ _tblname* Name of the table to create, check or modify

References `reset()`, and `_OWL::set_status()`.

6.26.3.8 SchemeHandler::create_table () [private]

Create the defined table

References `_define_field()`, and `scheme()`.

Referenced by `scheme()`.

6.26.3.9 SchemeHandler::define_index (\$ _index)

Define the indexes for a table

Parameters

$\leftarrow \$_index$ Array holding the index description. This is a 2 dimensional array where the first level holds the indexname. The second array defines the attributes for each index:

- `unique` : Boolean; True for unique keys
- `primary` : Boolean; True for the primary key
- `columns` : Array; List with columnnames that will be indexed
- `type` : String (optional); Index type, currently only supports 'FULLTEXT'

References `$_index`, `scheme()`, `_OWL::set_status()`, and `validate_scheme()`.

6.26.3.10 SchemeHandler::define_scheme (\$ _scheme)

Define the layout for a table

Parameters

$\leftarrow \$_scheme$ Array holding the table description. This is a 2 dimensional array where the first level holds the fieldnames. The second array defines the attributes for each field:

- `type` : String; the field-type (INT TINYINT VARCHAR MEDIUMTEXT TEXT LONGTEXT BLOB L
- `length` : Integer; indicating the length for fieldtypes that use that (like INT and VARCHAR)
- `null` : Boolean; when true the value can be NULL
- `auto-inc` : Boolean; True for auto-increment values (will be set as primary key)
- `default` : Mixed; default value
- `options` : Array; for SET and ENUM types. the list of possible values
- `comment` : String; field comment

References `$_scheme`, `scheme()`, `_OWL::set_status()`, and `validate_scheme()`.

6.26.3.11 static SchemeHandler::get_instance () [static]

Return a reference to my implementation. If necessary, create that implementation first.

Returns

Severity level

References `$instance`.

Referenced by `__construct()`.

6.26.3.12 _OWL::get_severity (\$ status = null) [inherited]

Get the current object severity level.

Parameters

← *\$status* An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References _OWL::\$status.

Referenced by LogHandler::compose_message(), LogHandler::log(), and DataHandler::set_key().

6.26.3.13 _OWL::get_status () [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by compare().

6.26.3.14 SchemeHandler::get_table_columns (\$ _tablename) [private]

Get the columns for a given table

Parameters

← *\$_tablename* The tablename

Returns

Indexed array holding all fields => datatypes, or null on errors

References _OWL::set_status().

Referenced by table_description().

6.26.3.15 SchemeHandler::get_table_indexes (\$ _tablename) [private]

Get the indexes for a given table

Parameters

← *\$_tablename* The tablename

Returns

Indexed array holding all fields => datatypes, or null on errors

References \$_index, and _OWL::set_status().

Referenced by table_description().

6.26.3.16 OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References [OWL::factory\(\)](#).

Referenced by [UserHandler::__construct\(\)](#), [SessionHandler::__construct\(\)](#), [__construct\(\)](#), [LogHandler::__construct\(\)](#), [ImageHandler::__construct\(\)](#), [FormHandler::__construct\(\)](#), [FileHandler::__construct\(\)](#), [DbHandler::__construct\(\)](#), [OWL::__construct\(\)](#), [Dispatcher::__construct\(\)](#), and [DataHandler::DataHandler\(\)](#).

6.26.3.17 SchemeHandler::reset ()

General reset function for all objects. Should be called after each non-fatal error

Reimplemented from [_OWL](#).

References [scheme\(\)](#).

Referenced by [create_scheme\(\)](#).

6.26.3.18 SchemeHandler::scheme (\$ _drops = false)

If the table does not exist, or differs from the defined scheme, create or modify the table

Parameters

$\leftarrow \$_drops$ True if existing fields should be dropped; default false. If existing fields should be converted to new fields, call with [DbScheme::scheme\(false\)](#) first, then do the conversions, next call [DbScheme::scheme\(true\)](#).

Returns

Severity level

References [alter_table\(\)](#), [compare\(\)](#), [create_table\(\)](#), and [_OWL::set_status\(\)](#).

Referenced by [alter_scheme\(\)](#), [compare\(\)](#), [create_table\(\)](#), [define_index\(\)](#), [define_scheme\(\)](#), [reset\(\)](#), and [validate_scheme\(\)](#).

6.26.3.19 OWL::set_high_severity (&\$ object = null) [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by [_OWL::check\(\)](#), [DataHandler::db\(\)](#), [DataHandler::prepare\(\)](#), and [SessionHandler::read\(\)](#).

6.26.3.20 OWL::set_status (\$ status, \$ params = array ()) [final, protected, inherited]

Set the current object status to the specified value.

Parameters

← **\$status** OWL status code
 ← **\$params**

References \$GLOBALS, _OWL::\$status, ConfigHandler::get(), Register::get_code(), _OWL::reset(), and _OWL::signal().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), __construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), Form::addField(), alter_scheme(), FileHandler::close(), DbHandler::connect(), DbHandler::create(), create_scheme(), DataHandler::DataHandler(), define_index(), define_scheme(), Dispatcher::dispatch(), FormHandler::get(), DataHandler::get(), get_table_columns(), get_table_indexes(), UserHandler::login(), FileHandler::open(), DbHandler::open(), LogHandler::open_logfile(), DataHandler::prepare(), DbHandler::prepare_delete(), DbHandler::prepare_insert(), DbHandler::prepare_read(), DbHandler::prepare_update(), DbHandler::read(), FileHandler::read_line(), UserHandler::read_userdata(), DataHandler::reset(), scheme(), FormHandler::set(), DataHandler::set(), DataHandler::set_join(), DataHandler::set_key(), Form::setFieldAttributes(), FormFieldText::setMaxsize(), FormFieldRadio::setSelected(), FormFieldText::setSize(), FormFieldSelect::setSize(), FormFieldSelect::setValue(), FormFieldRadio::setValue(), Form::showField(), table_description(), validate_scheme(), and DbHandler::write().

6.26.3.21 _OWL::signal (\$level = OWL_INFO, &\$text = false) [inherited]

Display the message for the current object status

Parameters

← **\$level** An optional severity level; message will only be displayed when it is at least of this level.
\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by _OWL::set_status(), and _OWL::traceback().

6.26.3.22 SchemeHandler::table_description (\$tablename, &\$data)

Get a description of a database table

Parameters

← **\$tablename** The tablename
\$data Indexed array holding all fields => datatypes

Returns

Severity level

References get_table_columns(), get_table_indexes(), and _OWL::set_status().

Referenced by compare().

6.26.3.23 _OWL::traceback (&\$ *text* = false, \$ *depth* = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References [_OWL::signal\(\)](#).

Referenced by [_OWL::check\(\)](#), [UserHandler::login\(\)](#), and [SessionHandler::read\(\)](#).

6.26.3.24 SchemeHandler::validate_scheme () [private]

Validate the defined scheme. Some values will be modified to make sure the SQL statements can be prepared and [compare\(\)](#) won't find differences on case diffs

Returns

boolean False if there is an error in the scheme definition, True if no errors were found

References [scheme\(\)](#), and [_OWL::set_status\(\)](#).

Referenced by [alter_scheme\(\)](#), [define_index\(\)](#), and [define_scheme\(\)](#).

6.26.4 Member Data Documentation

6.26.4.1 SchemeHandler::\$db [private]

integer - Reference to the database class

6.26.4.2 SchemeHandler::\$id [private]

integer - Scheme Handle ID

6.26.4.3 SchemeHandler::\$instance [static, private]

integer - self reference

Referenced by [get_instance\(\)](#).

6.26.4.4 SchemeHandler::\$inuse = false [private]

boolean - True when a scheme is filled with data and not yet used

6.26.4.5 SchemeHandler::\$scheme [private]

Array - table description

6.26.4.6 __OWL::\$severity [protected, inherited]

Severity level of the current object status

6.26.4.7 SchemeHandler::\$table = " " [private]

Array - table name

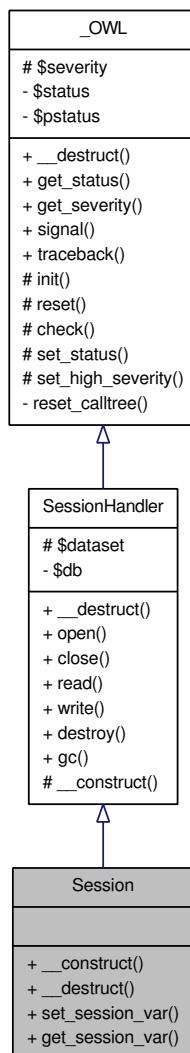
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/so/[class.schemehandler.php](#)

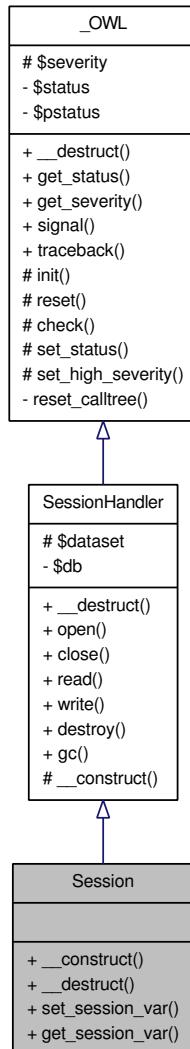
6.27 Session Class Reference

the OWL-PHP session object

Inheritance diagram for Session:



Collaboration diagram for Session:



Public Member Functions

- `__construct ()`
- `__destruct ()`
- `set_session_var ($var, $val=0, $flg=SESSIONVAR_SET)`
- `get_session_var ($var, $default=null)`
- `open ()`
- `close ()`
- `read ($id)`
- `write ($id, $data)`
- `destroy ($id)`
- `gc ($lifetime)`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Protected Member Functions

- `init ()`
- `reset ()`
- `check (&$object, $level)`
- `set_status ($status, $params=array())`
- `set_high_severity (&$object=null)`

Protected Attributes

- `$dataset`
- `$severity`

6.27.1 Detailed Description

the OWL-PHP session object This class handles the [OWL](#) session

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Aug 13, 2008 -- O van Eijk -- initial version

6.27.2 Constructor & Destructor Documentation

6.27.2.1 Session::__construct ()

When a new run is initialised, restore an older session or create a new one

Reimplemented from [SessionHandler](#).

6.27.2.2 Session::__destruct ()

When a run ends, write the sessiondata

Reimplemented from [SessionHandler](#).

6.27.3 Member Function Documentation

6.27.3.1 __OWL::check (&\$ object, \$ level) [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← `$object` Pointer to an object to check against
- ← `$level` The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.27.3.2 SessionHandler::close () [inherited]

Close the session

Returns

`bool`

6.27.3.3 SessionHandler::destroy (\$ id) [inherited]

Destroy a session

Parameters

← `$id` ID of the session to destroy

Returns

Boolean indicating success (true) or failure (false)

6.27.3.4 SessionHandler::gc (\$ lifetime) [inherited]

Garbage Collector. By default, this has a 1% chance of being executed (`session.gc_probability/session.gc_divisor`). The values can be changed in the constructor.

Parameters

← `$lifetime` `Session` lifetime in seconds. By default, 1440 seconds. This can be changed in the constructor

Returns

Status code

6.27.3.5 Session::get_session_var (\$ var, \$ default = null)

Get a session variable

Parameters

← `$var` Variable name

← `$default` Default value to return if the variable was not set (default null)

Returns

The value from the session, null if not set

6.27.3.6 OWL::get_severity (\$ *status* = null) [inherited]

Get the current object severity level.

Parameters

← *\$status* An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References OWL::\$status.

Referenced by LogHandler::compose_message(), LogHandler::log(), and DataHandler::set_key().

6.27.3.7 OWL::get_status () [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by SchemeHandler::compare().

6.27.3.8 OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References OWL::factory().

Referenced by UserHandler::__construct(), SessionHandler::__construct(), SchemeHandler::__construct(), LogHandler::__construct(), ImageHandler::__construct(), FormHandler::__construct(), FileHandler::__construct(), DbHandler::__construct(), OWL::__construct(), Dispatcher::__construct(), and DataHandler::DataHandler().

6.27.3.9 SessionHandler::open () [inherited]

Open the session

Returns

bool

6.27.3.10 SessionHandler::read (\$ *id*) [inherited]

Read the session

Parameters

← *\$id* session id

Returns

[Session](#) data

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

6.27.3.11 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References `_OWL::reset_calltree()`.

Referenced by `_OWL::check()`, `SessionHandler::read()`, `DataHandler::reset()`, and `_OWL::set_status()`.

6.27.3.12 `_OWL::set_high_severity (&$ object = null)` [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.27.3.13 `Session::set_session_var ($var, $val = 0, $flg = SESSIONVAR_SET)`

Set a session variable

Parameters

← `$var` Variable name

← `$val` Variable value (default 0)

← `$flg` How to handle the value. Default SESSIONVAR_SET

6.27.3.14 `_OWL::set_status ($status, $params = array ())` [final, protected, inherited]

Set the current object status to the specified value.

Parameters

← `$status` OWL status code

← `$params`

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler::DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`,

Dispatcher::dispatch(), FormHandler::get(), DataHandler::get(), SchemeHandler::get_table_columns(), SchemeHandler::get_table_indexes(), UserHandler::login(), FileHandler::open(), DbHandler::open(), LogHandler::open_logfile(), DataHandler::prepare(), DbHandler::prepare_delete(), DbHandler::prepare_insert(), DbHandler::prepare_read(), DbHandler::prepare_update(), DbHandler::read(), FileHandler::read_line(), UserHandler::read_userdata(), DataHandler::reset(), SchemeHandler::scheme(), FormHandler::set(), DataHandler::set(), DataHandler::set_join(), DataHandler::set_key(), Form::setFieldAttributes(), FormFieldText::setMaxsize(), FormFieldRadio::setSelected(), FormFieldText::setSize(), FormFieldSelect::setSize(), FormFieldSelect::setValue(), FormFieldRadio::setValue(), Form::showField(), SchemeHandler::table_description(), SchemeHandler::validate_scheme(), and DbHandler::write().

6.27.3.15 `_OWL::signal ($level = OWL_INFO, &$text = false)` [inherited]

Display the message for the current object status

Parameters

- ← **\$level** An optional severity level; message will only be displayed when it is at least of this level.
- ← **\$text** If this parameter is given, the message text is returned in this string instead of echo'd.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by _OWL::set_status(), and _OWL::traceback().

6.27.3.16 `_OWL::traceback (&$text = false, $depth = 0)` [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

- \$text** Optional variable in which the message text can be stored. If not given, the text will be written to standard output
- ← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References _OWL::signal().

Referenced by _OWL::check(), UserHandler::login(), and SessionHandler::read().

6.27.3.17 `SessionHandler::write ($id, $data)` [inherited]

Write the session

Parameters

← *\$id* Session id
← *\$data* Session data

Returns

True on success, Failure otherwise

References `_OWL::check()`.

6.27.4 Member Data Documentation

6.27.4.1 SessionHandler::\$dataset [protected, inherited]

Link to a datahandler object. This dataset is used as an interface to all database IO.

6.27.4.2 `_OWL::$severity` [protected, inherited]

Severity level of the current object status

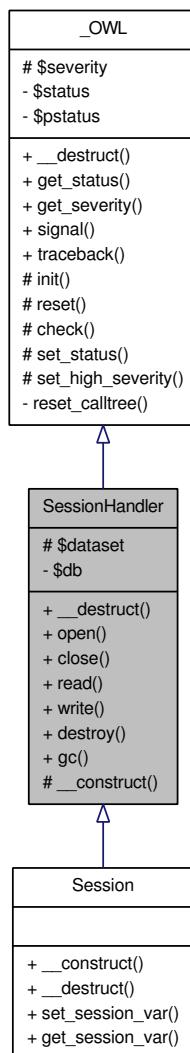
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/bo/[class.session.php](#)

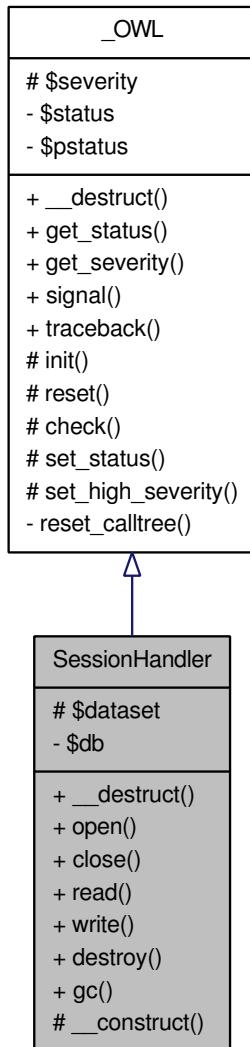
6.28 SessionHandler Class Reference

the PHP session object

Inheritance diagram for SessionHandler:



Collaboration diagram for SessionHandler:



Public Member Functions

- `__destruct ()`
- `open ()`
- `close ()`
- `read ($id)`
- `write ($id, $data)`
- `destroy ($id)`
- `gc ($lifetime)`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Protected Member Functions

- `__construct ()`
- `init ()`
- `reset ()`
- `check (&$object, $level)`
- `set_status ($status, $params=array())`
- `set_high_severity (&$object=null)`

Protected Attributes

- `$dataset`
- `$severity`

Private Attributes

- `$db`

6.28.1 Detailed Description

the PHP session object This class saves and (re)stores the user sessions

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Jul 29, 2008 -- O van Eijk -- initial version

6.28.2 Constructor & Destructor Documentation

6.28.2.1 SessionHandler::__construct () [protected]

Class constructor; set the save_handler

Reimplemented in [Session](#).

References `OWL::factory()`, `ConfigHandler::get()`, `_OWL::init()`, and `_OWL::set_status()`.

6.28.2.2 SessionHandler::__destruct ()

Write the session data back to the database

Reimplemented from [_OWL](#).

Reimplemented in [Session](#).

6.28.3 Member Function Documentation

6.28.3.1 `_OWL::check (&$ object, $ level)` [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← **\$object** Pointer to an object to check against
- ← **\$level** The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `write()`.

6.28.3.2 `SessionHandler::close ()`

Close the session

Returns

bool

6.28.3.3 `SessionHandler::destroy ($ id)`

Destroy a session

Parameters

- ← **\$id** ID of the session to destroy

Returns

Boolean indicating success (true) or failure (false)

6.28.3.4 `SessionHandler::gc ($ lifetime)`

Garbage Collector. By default, this has a 1% chance of being executed (`session.gc_probability/session.gc_divisor`). The values can be changed in the constructor.

Parameters

- ← **\$lifetime** Session lifetime in seconds. By default, 1440 seconds. This can be changed in the constructor

Returns

Status code

6.28.3.5 `_OWL::get_severity ($status = null)` [inherited]

Get the current object severity level.

Parameters

← `$status` An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References `_OWL::$status`.

Referenced by `LogHandler::compose_message()`, `LogHandler::log()`, and `DataHandler::set_key()`.

6.28.3.6 `_OWL::get_status ()` [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by `SchemeHandler::compare()`.

6.28.3.7 `_OWL::init ()` [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References `OWL::factory()`.

Referenced by `UserHandler::__construct()`, `__construct()`, `SchemeHandler::__construct()`, `LogHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `OWL::__construct()`, `Dispatcher::__construct()`, and `DataHandler::DataHandler()`.

6.28.3.8 `SessionHandler::open ()`

Open the session

Returns

bool

6.28.3.9 `SessionHandler::read ($id)`

Read the session

Parameters

← `$id` session id

Returns

[Session](#) data

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

6.28.3.10 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References `_OWL::reset_calltree()`.

Referenced by `_OWL::check()`, `read()`, `DataHandler::reset()`, and `_OWL::set_status()`.

6.28.3.11 `_OWL::set_high_severity (&$ object = null) [protected, inherited]`

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `read()`.

6.28.3.12 `_OWL::set_status ($status, $params = array ()) [final, protected, inherited]`

Set the current object status to the specified value.

Parameters

← `$status` [OWL](#) status code

← `$params`

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `UserHandler::__construct()`, `__construct()`, `SchemeHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler::DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`, `Dispatcher::dispatch()`, `FormHandler::get()`, `DataHandler::get()`, `SchemeHandler::get_table_columns()`, `SchemeHandler::get_table_indexes()`, `UserHandler::login()`, `FileHandler::open()`, `DbHandler::open()`, `LogHandler::open_logfile()`, `DataHandler::prepare()`, `DbHandler::prepare_delete()`, `DbHandler::prepare_insert()`, `DbHandler::prepare_read()`, `DbHandler::prepare_update()`, `DbHandler::read()`, `FileHandler::read_line()`, `UserHandler::read_userdata()`, `DataHandler::reset()`, `SchemeHandler::scheme()`, `FormHandler::set()`, `DataHandler::set()`, `DataHandler::set_join()`, `DataHandler::set_key()`, `Form::setFieldAttributes()`, `FormFieldText::setMaxsize()`, `FormFieldRadio::setSelected()`, `FormFieldText::setSize()`, `FormFieldSelect::setSize()`, `FormFieldSelect::setValue()`, `FormFieldRadio::setValue()`, `Form::showField()`, `SchemeHandler::table_description()`, `SchemeHandler::validate_scheme()`, and `DbHandler::write()`.

6.28.3.13 _OWL::signal (\$ level = OWL_INFO, &\$ text = false) [inherited]

Display the message for the current object status

Parameters

← **\$level** An optional severity level; message will only be displayed when it is at least of this level.

← **\$text** If this parameter is given, the message text is returned in this string instead of echo'd.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by _OWL::set_status(), and _OWL::traceback().

6.28.3.14 _OWL::traceback (&\$ text = false, \$ depth = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

← **\$text** Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References _OWL::signal().

Referenced by _OWL::check(), UserHandler::login(), and read().

6.28.3.15 SessionHandler::write (\$ id, \$ data)

Write the session

Parameters

← **\$id** Session id

← **\$data** Session data

Returns

True on success, Failure otherwise

References _OWL::check().

6.28.4 Member Data Documentation

6.28.4.1 SessionHandler::\$dataset [protected]

Link to a datahandler object. This dataset is used as an interface to all database IO.

6.28.4.2 SessionHandler::\$db [private]

Reference to the DB Singleton

6.28.4.3 _OWL::\$severity [protected, inherited]

Severity level of the current object status

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

- /home/oscar/projects/owl-php/src/kernel/so/[class.sessionhandler.php](#)

6.29 StatusHandler Class Reference

Status object.

Public Member Functions

- `__clone ()`
- `set_code ($code=OWL_STATUS_BUG)`
- `reset ()`
- `set_params ($params)`
- `get_severity ($status=null)`
- `get_code ()`
- `get_message ()`

Static Public Member Functions

- static `get_instance ()`

Private Member Functions

- `__construct ($code=OWL_STATUS_WARNING)`

Private Attributes

- `$code`
- `$params`

Static Private Attributes

- static `$instance`

6.29.1 Detailed Description

Status object. Each object, when initialised, gets a status object which holds information about the last action that was performed.

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Aug 11, 2008 -- O van Eijk -- Initial version

6.29.2 Constructor & Destructor Documentation

6.29.2.1 StatusHandler::__construct (\$code = OWL_STATUS_WARNING) [private]

Constructor; should be called only by [_OWL::init\(\)](#). The default status is initially a (generic) warning status. It should be set to any successfull status after object initialisation completed.

Parameters

← **\$code** The status code

References \$code.

6.29.3 Member Function Documentation

6.29.3.1 StatusHandler::__clone ()

Implementation of the [__clone\(\)](#) function to prevent cloning of this singleton; it triggers a fatal (user)error

6.29.3.2 StatusHandler::get_code ()

Return the status code

Returns

The status code

6.29.3.3 static StatusHandler::get_instance () [static]

Return a reference to my implementation. If necessary, create that implementation first.

Returns

Severity level

References \$instance.

6.29.3.4 StatusHandler::get_message ()

Retrieve the message that has been specified for the current object status. If parameters are set and wanted by the message text, replace them. If more parameters are available then used in the message, there's a quote with the unused parameter count added to the message.

Returns

The message text

References \$GLOBALS, Register::get_code(), and Register::register_messages().

6.29.3.5 StatusHandler::get_severity (\$status = null)

Check the status of the given object and return its severity.

Parameters

← *\$status* An optional parameter to check an other status code i.s.o the object's current status.

Returns

The severity level of the current status

6.29.3.6 StatusHandler::reset ()

Reset the object status

6.29.3.7 StatusHandler::set_code (\$code = OWL_STATUS_BUG)

Set the status of the owner object to the given value.

Parameters

← *\$code* The status code

Returns

The severity level of the code

References \$code.

6.29.3.8 StatusHandler::set_params (\$params)

If the status was set with optional parameters, they will be set in this subject and substituted in the correct message

Parameters

← *\$params* An array with parameters

References \$params.

6.29.4 Member Data Documentation**6.29.4.1 StatusHandler::\$code [private]**

Current object status

Referenced by __construct(), and set_code().

6.29.4.2 StatusHandler::\$instance [static, private]

integer - self reference

Referenced by get_instance().

6.29.4.3 StatusHandler::\$params [private]

Array with parameters that will be substituted in message text

Referenced by `set_params()`.

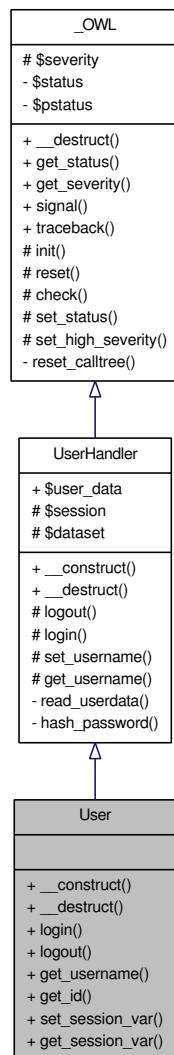
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/so/[class.statushandler.php](#)

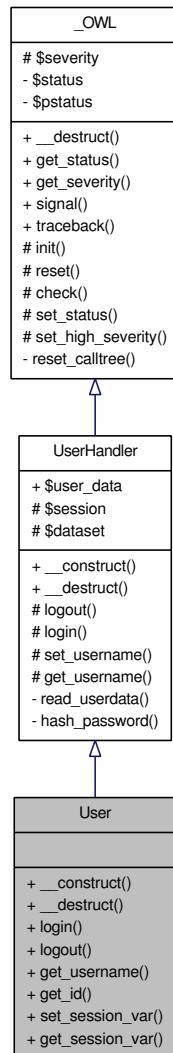
6.30 User Class Reference

the OWL-PHP session object

Inheritance diagram for User:



Collaboration diagram for User:



Public Member Functions

- `__construct ($username=null)`
- `__destruct ()`
- `login ($username, $password)`
- `logout ()`
- `get_username ()`
- `get_id ()`
- `set_session_var ($var, $val=0, $flg=SESSIONVAR_SET)`
- `get_session_var ($var, $default=null)`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Public Attributes

- `$user_data`

Protected Member Functions

- `set_username` (`$username`)
- `init` ()
- `reset` ()
- `check` (`&$object, $level`)
- `set_status` (`$status, $params=array()`)
- `set_high_severity` (`&$object=null`)

Protected Attributes

- `$session`
- `$dataset`
- `$severity`

6.30.1 Detailed Description

the OWL-PHP session object This class handles the [OWL](#) users

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Aug 27, 2008 -- O van Eijk -- initial version

6.30.2 Constructor & Destructor Documentation

6.30.2.1 `User::__construct ($username = null)`

Class constructor; create a new user environment

Parameters

`← $username` Username when logged in. Default username is 'anonymous'

Reimplemented from [UserHandler](#).

References `ConfigHandler::get()`.

6.30.2.2 `User::__destruct ()`

When a run ends, write the sessiondata

Reimplemented from [UserHandler](#).

6.30.3 Member Function Documentation

6.30.3.1 `_OWL::check (&$ object, $ level)` [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

- ← `$object` Pointer to an object to check against
- ← `$level` The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References `_OWL::reset()`, `_OWL::set_high_severity()`, and `_OWL::traceback()`.

Referenced by `SessionHandler::write()`.

6.30.3.2 `User::get_id ()`

Return the current session ID

Returns

the session ID

6.30.3.3 `User::get_session_var ($ var, $ default = null)`

Get a session variable

Parameters

- ← `$var` Variable name
- ← `$default` Default value to return if the variable was not set (default null)

Returns

The value from the session, null if not set

6.30.3.4 `_OWL::get_severity ($ status = null)` [inherited]

Get the current object severity level.

Parameters

- ← `$status` An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References `_OWL::$status`.

Referenced by `LogHandler::compose_message()`, `LogHandler::log()`, and `DataHandler::set_key()`.

6.30.3.5 `_OWL::get_status () [final, inherited]`

Get the current object status.

Returns

Object's status code

Referenced by [SchemeHandler::compare\(\)](#).

6.30.3.6 `User::get_username ()`

Return the username of the current session

Reimplemented from [UserHandler](#).

6.30.3.7 `_OWL::init () [protected, inherited]`

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References [OWL::factory\(\)](#).

Referenced by [UserHandler::__construct\(\)](#), [SessionHandler::__construct\(\)](#), [SchemeHandler::__construct\(\)](#), [LogHandler::__construct\(\)](#), [ImageHandler::__construct\(\)](#), [FormHandler::__construct\(\)](#), [FileHandler::__construct\(\)](#), [DbHandler::__construct\(\)](#), [OWL::__construct\(\)](#), [Dispatcher::__construct\(\)](#), and [DataHandler::DataHandler\(\)](#).

6.30.3.8 `User::login ($username, $password)`

Log in

Parameters

- ← **\$username** Given username
- ← **\$password** Given password

Returns

True on success, False otherwise

Reimplemented from [UserHandler](#).

References [UserHandler::set_username\(\)](#).

6.30.3.9 `User::logout ()`

Log out the current user Note: After logging out, the session still continues. The calling app must take care of the forward (e.g. with a header('location: ' . \$_SERVER['PHP_SELF'])) after a call to [User::logout\(\)](#).

Reimplemented from [UserHandler](#).

References [ConfigHandler::get\(\)](#), and [UserHandler::set_username\(\)](#).

6.30.3.10 `_OWL::reset ()` [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References `_OWL::reset_calltree()`.

Referenced by `_OWL::check()`, `SessionHandler::read()`, `DataHandler::reset()`, and `_OWL::set_status()`.

6.30.3.11 `_OWL::set_high_severity (&$ object = null)` [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.30.3.12 `User::set_session_var ($var, $val = 0, $flg = SESSIONVAR_SET)`

Set a session variable

Parameters

$\leftarrow \$var$ Variable name

$\leftarrow \$val$ Variable value (default 0)

$\leftarrow \$flg$ How to handle the value. Default SESSIONVAR_SET

6.30.3.13 `_OWL::set_status ($status, $params = array ())` [final, protected, inherited]

Set the current object status to the specified value.

Parameters

$\leftarrow \$status$ [OWL](#) status code

$\leftarrow \$params$

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `UserHandler::__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler::DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`, `Dispatcher::dispatch()`, `FormHandler::get()`, `DataHandler::get()`, `SchemeHandler::get_table_columns()`, `SchemeHandler::get_table_indexes()`, `UserHandler::login()`, `FileHandler::open()`, `DbHandler::open()`, `LogHandler::open_logfile()`, `DataHandler::prepare()`, `DbHandler::prepare_delete()`, `DbHandler::prepare_insert()`, `DbHandler::prepare_read()`, `DbHandler::prepare_update()`, `DbHandler::read()`, `FileHandler::read_line()`, `UserHandler::read_userdata()`, `DataHandler::reset()`, `SchemeHandler::scheme()`, `FormHandler::set()`, `DataHandler::set()`,

DataHandler::set_join(), DataHandler::set_key(), Form::setFieldAttributes(), FormFieldText::setMaxsize(), FormFieldRadio::setSelected(), FormFieldText::setSize(), FormFieldSelect::setSize(), FormFieldSelect::setValue(), FormFieldRadio::setValue(), Form::showField(), SchemeHandler::table_description(), SchemeHandler::validate_scheme(), and DbHandler::write().

6.30.3.14 UserHandler::set_username (\$username) [protected, inherited]

Set the username

Parameters

← **\$username** Username

Referenced by UserHandler::__construct(), UserHandler::login(), login(), and logout().

6.30.3.15 _OWL::signal (\$level = OWL_INFO, &\$text = false) [inherited]

Display the message for the current object status

Parameters

← **\$level** An optional severity level; message will only be displayed when it is at least of this level.

\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References ConfigHandler::get().

Referenced by _OWL::set_status(), and _OWL::traceback().

6.30.3.16 _OWL::traceback (&\$text = false, \$depth = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References _OWL::signal().

Referenced by _OWL::check(), UserHandler::login(), and SessionHandler::read().

6.30.4 Member Data Documentation

6.30.4.1 `UserHandler::$dataset` [protected, inherited]

Link to a datahandler object. This dataset is used as an interface to all database IO.

6.30.4.2 `UserHandler::$session` [protected, inherited]

The PHP session object

6.30.4.3 `_OWL::$severity` [protected, inherited]

Severity level of the current object status

6.30.4.4 `UserHandler::$user_data` [inherited]

An indexed array with the user information as take from the database

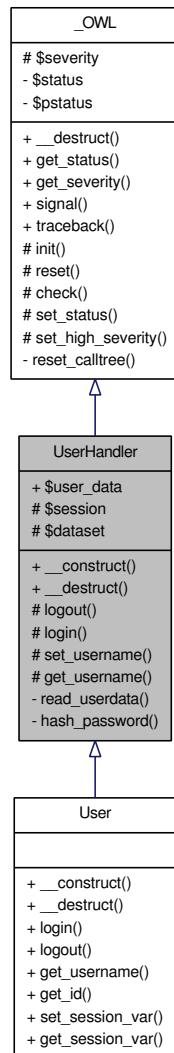
The documentation for this class was generated from the following file:

- /home/oscar/projects/owl-php/src/kernel/bo/[class.user.php](#)

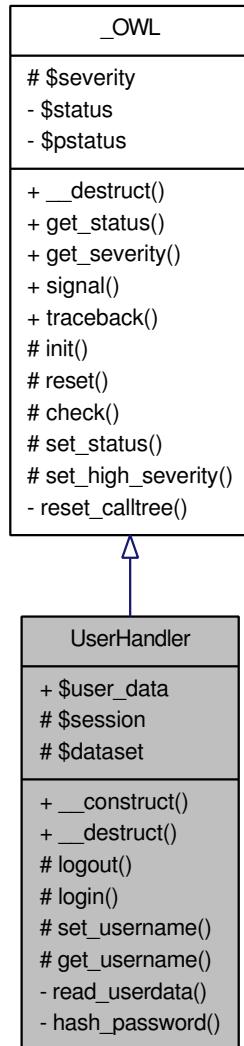
6.31 UserHandler Class Reference

the user object

Inheritance diagram for UserHandler:



Collaboration diagram for UserHandler:



Public Member Functions

- `__construct ($username)`
- `__destruct ()`
- `get_status ()`
- `get_severity ($status=null)`
- `signal ($level=OWL_INFO, &$text=false)`
- `traceback (&$text=false, $depth=0)`

Public Attributes

- `$user_data`

Protected Member Functions

- `logout ()`
- `login ($username, $password)`
- `set_username ($username)`
- `get_username ()`
- `init ()`
- `reset ()`
- `check (&$object, $level)`
- `set_status ($status, $params=array())`
- `set_high_severity (&$object=null)`

Protected Attributes

- `$session`
- `$dataset`
- `$severity`

Private Member Functions

- `read_userdata ()`
- `hash_password ($password)`

6.31.1 Detailed Description

the user object This class creates user sessions and handles logging in and out

Author

Oscar van Eijk, Oveas Functionality Provider

Version

Aug 27, 2008 -- O van Eijk -- initial version

6.31.2 Constructor & Destructor Documentation

6.31.2.1 UserHandler::__construct (\$username)

Class constructor; setup the user environment

Parameters

$\leftarrow \$username$ Username.

Reimplemented in [User](#).

References `_OWL::init()`, `read_userdata()`, `_OWL::set_status()`, and `set_username()`.

6.31.2.2 UserHandler::__destruct ()

Class destructor

Reimplemented from [_OWL](#).

Reimplemented in [User](#).

6.31.3 Member Function Documentation

6.31.3.1 _OWL::check (&\$ object, \$ level) [protected, inherited]

This is a helper function for lazy developers. Some checks have to be made quite often, this is a kinda macro to handle that. It compares the own severity level with that of a given object. If the highest level is above a given max, a traceback and reset are performed.

Parameters

← *\$object* Pointer to an object to check against

← *\$level* The maximum severity level

Returns

True if the severity level was correct (below the max), otherwise false

References [_OWL::reset\(\)](#), [_OWL::set_high_severity\(\)](#), and [_OWL::traceback\(\)](#).

Referenced by [SessionHandler::write\(\)](#).

6.31.3.2 _OWL::get_severity (\$ status = null) [inherited]

Get the current object severity level.

Parameters

← *\$status* An optional parameter to check an other status code i.s.o the object's current status.

Returns

Status severity level

References [_OWL::\\$status](#).

Referenced by [LogHandler::compose_message\(\)](#), [LogHandler::log\(\)](#), and [DataHandler::set_key\(\)](#).

6.31.3.3 _OWL::get_status () [final, inherited]

Get the current object status.

Returns

Object's status code

Referenced by [SchemeHandler::compare\(\)](#).

6.31.3.4 UserHandler::get_username () [protected]

Return the username of the current session

Reimplemented in [User](#).

6.31.3.5 UserHandler::hash_password (\$ password) [private]

Encrypt a given password

Parameters

← **\$password** Given password in plain text format

Returns

The encrypted password

References [ConfigHandler::get\(\)](#).

Referenced by [login\(\)](#).

6.31.3.6 __OWL::init () [protected, inherited]

This function should be called by all constructors. It initializes the general characteristics. Status is 'warning' by default, it's up to the constructor to set a proper status; if it's still 'warning', this *might* indicate something went wrong.

References [OWL::factory\(\)](#).

Referenced by [__construct\(\)](#), [SessionHandler::__construct\(\)](#), [SchemeHandler::__construct\(\)](#), [LogHandler::__construct\(\)](#), [ImageHandler::__construct\(\)](#), [FormHandler::__construct\(\)](#), [FileHandler::__construct\(\)](#), [DbHandler::__construct\(\)](#), [OWL::__construct\(\)](#), [Dispatcher::__construct\(\)](#), and [DataHandler::DataHandler\(\)](#).

6.31.3.7 UserHandler::login (\$ username, \$ password) [protected]

Attempt to log in with the current user and the given password

Parameters

← **\$username** Given username. Might be taken from the session as well, but given as a parameter here to suppress the E_STRICT Declaration warning

← **\$password** The user provided password

Returns

True on success, False otherwise

Reimplemented in [User](#).

References [ConfigHandler::get\(\)](#), [hash_password\(\)](#), [__OWL::set_status\(\)](#), [set_username\(\)](#), and [__OWL::traceback\(\)](#).

6.31.3.8 UserHandler::logout () [protected]

Log out the current user.

Reimplemented in [User](#).

6.31.3.9 UserHandler::read_userdata () [private]

When a new session starts for a use that was logged in before retrieve the userdata back from the database

References `_OWL::set_status()`.

Referenced by `__construct()`.

6.31.3.10 _OWL::reset () [protected, inherited]

General reset function for all objects. Should be called after each non-fatal error

Reimplemented in [DbHandler](#), and [SchemeHandler](#).

References `_OWL::reset_calltree()`.

Referenced by `_OWL::check()`, `SessionHandler::read()`, `DataHandler::reset()`, and `_OWL::set_status()`.

6.31.3.11 _OWL::set_high_severity (&\$ object = null) [protected, inherited]

Compare the severity level of the current object with a given one and set my statuspointer to the object with the highest level.

Referenced by `_OWL::check()`, `DataHandler::db()`, `DataHandler::prepare()`, and `SessionHandler::read()`.

6.31.3.12 _OWL::set_status (\$ status, \$ params = array ()) [final, protected, inherited]

Set the current object status to the specified value.

Parameters

$\leftarrow \$status$ OWL status code

$\leftarrow \$params$

References `$GLOBALS`, `_OWL::$status`, `ConfigHandler::get()`, `Register::get_code()`, `_OWL::reset()`, and `_OWL::signal()`.

Referenced by `__construct()`, `SessionHandler::__construct()`, `SchemeHandler::__construct()`, `ImageHandler::__construct()`, `FormHandler::__construct()`, `FileHandler::__construct()`, `DbHandler::__construct()`, `Form::addField()`, `SchemeHandler::alter_scheme()`, `FileHandler::close()`, `DbHandler::connect()`, `DbHandler::create()`, `SchemeHandler::create_scheme()`, `DataHandler::DataHandler()`, `SchemeHandler::define_index()`, `SchemeHandler::define_scheme()`, `Dispatcher::dispatch()`, `FormHandler::get()`, `DataHandler::get()`, `SchemeHandler::get_table_columns()`, `SchemeHandler::get_table_indexes()`, `login()`, `FileHandler::open()`, `DbHandler::open()`, `LogHandler::open_logfile()`, `DataHandler::prepare()`, `DbHandler::prepare_`

`delete()`, `DbHandler::prepare_insert()`, `DbHandler::prepare_read()`, `DbHandler::prepare_update()`, `DbHandler::read()`, `FileHandler::read_line()`, `read_userdata()`, `DataHandler::reset()`, `SchemeHandler::scheme()`, `FormHandler::set()`, `DataHandler::set()`, `DataHandler::set_join()`, `DataHandler::set_key()`, `Form::setFieldAttributes()`, `FormFieldText::setMaxsize()`, `FormFieldRadio:: setSelected()`, `FormFieldText::setSize()`, `FormFieldSelect::setSize()`, `FormFieldSelect::setValue()`, `FormFieldRadio::setValue()`, `Form::showField()`, `SchemeHandler::table_description()`, `SchemeHandler::validate_scheme()`, and `DbHandler::write()`.

6.31.3.13 UserHandler::set_username (\$username) [protected]

Set the username

Parameters

← **\$username** Username

Referenced by `__construct()`, `login()`, `User::login()`, and `User::logout()`.

6.31.3.14 _OWL::signal (\$level = OWL_INFO, &\$text = false) [inherited]

Display the message for the current object status

Parameters

← **\$level** An optional severity level; message will only be displayed when it is at least of this level.

\$text If this parameter is given, the message text is returned in this string instead of echoed.

Returns

The severity level for this object

References `ConfigHandler::get()`.

Referenced by `_OWL::set_status()`, and `_OWL::traceback()`.

6.31.3.15 _OWL::traceback (&\$text = false, \$depth = 0) [inherited]

If somewhere in the nested calls an error occurred, we can traceback the original failing object with this function and signal the message.

Parameters

\$text Optional variable in which the message text can be stored. If not given, the text will be written to standard output

← **\$depth** This parameter should be initially empty. It calculates the depth in recursive calls.

Returns

Severity code of the failing object

References `_OWL::signal()`.

Referenced by `_OWL::check()`, `login()`, and `SessionHandler::read()`.

6.31.4 Member Data Documentation

6.31.4.1 UserHandler::\$dataset [protected]

Link to a datahandler object. This dataset is used as an interface to all database IO.

6.31.4.2 UserHandler::\$session [protected]

The PHP session object

6.31.4.3 _OWL::\$severity [protected, inherited]

Severity level of the current object status

6.31.4.4 UserHandler::\$user_data

An indexed array with the user information as take from the database

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

- /home/oscar/projects/owl-php/src/kernel/so/[class.userhandler.php](#)

Chapter 7

File Documentation

7.1 /home/oscar/projects/owl-php/src/config.php File Reference

Variables

- `$GLOBALS['register']`
- `$GLOBALS['config']['configfiles']['owl'] = OWL_ROOT . '/owl_config.cfg'`
- `$GLOBALS['config']['hide']['tag'] = '(hide)'`
- `$GLOBALS['config']['hide']['value'] = '(hidden)'`

7.1.1 Detailed Description

Configuration basics file for **OWL**; it stores some fixed configuration in a global data structure

Version

Id

[config.php](#), v 1.5 2010-08-20 08:39:54 oscar Exp

7.1.2 Variable Documentation

7.1.2.1 `$GLOBALS['config']['hide']['value'] = '(hidden)'`

7.1.2.2 `$GLOBALS['config']['hide']['tag'] = '(hide)'`

7.1.2.3 `$GLOBALS['config']['configfiles']['owl'] = OWL_ROOT . '/owl_config.cfg'`

7.1.2.4 `$GLOBALS['register']`

Initial value:

```
array (
    'applications'          => array()
```

```
        , 'classes'          => array(
    )
)
```

Referenced by OWLloader::tryLoad(), ConfigHandler::get(), Register::get_code(), StatusHandler::get_message(), Register::get_run_id(), Register::get_severity(), Register::get_severity_level(), Register::init(), OWLErrorHandler::log_exception(), ConfigHandler::read_config(), Register::register_app(), Register::register_class(), Register::register_code(), Register::register_severity(), ConfigHandler::set(), Register::set_application(), Register::set_class(), Register::set_severity(), and _OWL::set_status().

7.2 /home/oscar/projects/owl-php/src/index.php File Reference

Enumerations

- enum **OWL_ROOT**

Variables

- **if(!OWLloader::getClass('form')) \$LoginForm = new Form('applic#/include-path#/classfile#/class#/method')**
- **\$_form = OWL::factory('FormHandler')**
- **\$_user = new User()**
- **switch (\$_form->get('act'))**
- **\$_scheme = SchemeHandler::get_instance()**
- **\$_table**
- **\$_index**
- **\$_t = serialize(\$_data['columns'])**

7.2.1 Detailed Description

This is the entry point for OWL-PHP teststub

Version

Id

[index.php,v](#) 1.11 2010-12-03 12:07:43 oscar Exp

7.2.2 Enumeration Type Documentation

7.2.2.1 enum **OWL_ROOT**

7.2.3 Variable Documentation

7.2.3.1 **\$_form = OWL::factory('FormHandler')**

Referenced by Dispatcher::dispatch().

7.2.3.2 **\$_index**

Initial value:

```
array (
    'name' => array(
        'columns' => array ('name')
        , 'primary' => false
        , 'unique' => false
        , 'type' => null
    )
)
```

```

        , 'address' => array(
            'columns' => array ('address')
            , 'primary' => false
            , 'unique' => false
            , 'type' => 'FULLTEXT'
        )
    )
)

```

Referenced by SchemeHandler::define_index(), and SchemeHandler::get_table_indexes().

7.2.3.3 \$_scheme = SchemeHandler::get_instance()

Referenced by SchemeHandler::define_scheme().

7.2.3.4 \$_t = serialize(\$_data['columns'])

Referenced by DbHandler::expand_field(), DbHandler::extract_tablelist(), and DataHandler::prepare().

7.2.3.5 \$_table

Referenced by DbHandler::extract_tablelist(), DataHandler::prepare(), and DbHandler::tablelist().

7.2.3.6 \$_user = new User()

7.2.3.7 if (!OWLloader::getClass('form')) \$LoginForm = new Form('applic#include-path#classfile#class#method')

7.2.3.8 switch(\$_form->get('act'))

7.3 /home/oscar/projects/owl-php/src/kernel/bo/class.dispatcher.php File Reference

Classes

- class [Dispatcher](#)
Dispatcher singleton.

7.3.1 Detailed Description

This file defines the Oveas Web Library [Dispatcher](#) class

Version

Id

[class.dispatcher.php](#),v 1.1 2010-12-03 12:07:42 oscar Exp

7.4 /home/oscar/projects/owl-php/src/kernel/bo/class.form.php

File Reference

Classes

- class [Form](#)
Form Element class.

7.4.1 Detailed Description

This file defines the HTML [Form](#) class

Version

Id

[class.form.php](#),v 1.2 2010-12-03 12:07:42 oscar Exp

7.5 /home/oscar/projects/owl-php/src/kernel/bo/class.owl.php File Reference

Classes

- class [OWL](#)

7.5.1 Detailed Description

This file defines the Oveas Web Library helper class

Version

Id

[class.owl.php](#),v 1.2 2010-12-03 12:07:42 oscar Exp

7.6 /home/oscar/projects/owl-php/src/kernel/bo/class.session.php

File Reference

Classes

- class [Session](#)
the OWL-PHP session object

7.6.1 Detailed Description

This file defines the [Session](#) class

Version

Id

[class.session.php](#),v 1.5 2010-12-03 12:07:42 oscar Exp

7.7 /home/oscar/projects/owl-php/src/kernel/bo/class.user.php File Reference

Classes

- class [User](#)
the OWL-PHP session object

7.7.1 Detailed Description

This file defines the [User](#) class

Version

Id

[class.user.php](#),v 1.5 2010-12-03 12:07:42 oscar Exp

7.8 /home/oscar/projects/owl-php/src/kernel/class._-owl.php File Reference

Classes

- class [_OWL](#)

7.8.1 Detailed Description

This file defines the Oveas Web Library main class

Version

Id

[class._owl.php](#),v 1.2 2010-12-03 12:07:42 oscar Exp

7.9 /home/oscar/projects/owl-php/src/kernel/so/class.confighandler.php File Reference

Classes

- class [ConfigHandler](#)
Configuration handler.

7.9.1 Detailed Description

Define a class for config handling

Version

Id

[class.confighandler.php,v](#) 1.5 2010-12-03 12:07:42 oscar Exp

7.10 /home/oscar/projects/owl-php/src/kernel/so/class.datahandler.php File Reference

Classes

- class [DataHandler](#)

The OWL Data object.

Enumerations

Query preparation tools

These flags define what type of queries can be prepared

- enum [DATA_UNPREPARED](#)
Default value; no query prepared yet.
- enum [DATA_READ](#)
Read data from the database.
- enum [DATA_WRITE](#)
Write new data to the database.
- enum [DATA_UPDATE](#)
Update data in the database.
- enum [DATA_DELETE](#)
Remove data from the database.

Reset flags

These flags how an object should be performed. All values includes all lower values as well!

- enum [DATA_RESET_STATUS](#)
Reset object status only.
- enum [DATA_RESET_PREPARE](#)
Reset prepared queries as well.
- enum [DATA_RESET_META](#)
Remove all locks and joins.
- enum [DATA_RESET_FULL](#)
Clean all data.

7.10.1 Detailed Description

This file defines the [DataHandler](#) class

Version

Id

[class.datahandler.php](#), v 1.5 2010-12-03 12:07:42 oscar Exp

7.10.2 Enumeration Type Documentation

7.10.2.1 enum DATA_DELETE

Remove data from the database.

7.10.2.2 enum DATA_READ

Read data from the database.

7.10.2.3 enum DATA_RESET_FULL

Clean all data.

7.10.2.4 enum DATA_RESET_META

Remove all locks and joins.

7.10.2.5 enum DATA_RESET_PREPARE

Reset prepared queries as well.

7.10.2.6 enum DATA_RESET_STATUS

Reset object status only.

7.10.2.7 enum DATA_UNPREPARED

Default value; no query prepared yet.

7.10.2.8 enum DATA_UPDATE

Update data in the database.

7.10.2.9 enum DATA_WRITE

Write new data to the database.

7.11 /home/oscar/projects/owl-php/src/kernel/so/class.dbhandler.php File Reference

Classes

- class [DbHandler](#)

Database handler.

Enumerations

Return Flags

These flags that define what value should be returned by read()

- enum [DBHANDLE_DATA](#)

Return the read values (default).

- enum [DBHANDLE_SINGLEFIELD](#)

Return the read value as a single field (i.s.o. a 2D array).

- enum [DBHANDLE_SINGLEROW](#)

Return the read value as a single row (a 1D array).

- enum [DBHANDLE_ROWCOUNT](#)

Return the number of rows.

- enum [DBHANDLE_FIELDCOUNT](#)

Return the number of fields per row.

- enum [DBHANDLE_TOTALFIELDCOUNT](#)

Return the total number of fields.

7.11.1 Detailed Description

This file defines the Database Handler class

Version

Id

[class.dbhandler.php](#),v 1.8 2010-12-03 12:07:42 oscar Exp

7.11.2 Enumeration Type Documentation

7.11.2.1 enum DBHANDLE_DATA

Return the read values (default).

7.11.2.2 enum DBHANDLE_FIELDCOUNT

Return the number of fields per row.

7.11.2.3 enum DBHANDLE_ROWCOUNT

Return the number of rows.

7.11.2.4 enum DBHANDLE_SINGLEFIELD

Return the read value as a single field (i.s.o. a 2D array).

7.11.2.5 enum DBHANDLE_SINGLEROW

Return the read value as a single row (a 1D array).

7.11.2.6 enum DBHANDLE_TOTALFIELDCOUNT

Return the total number of fields.

7.12 /home/oscar/projects/owl-php/src/kernel/so/class.exceptionhandler File Reference

Classes

- class [OWLException](#)
Exception handler.
- class [OWLExceptionHandler](#)
Default exception handler.

7.12.1 Detailed Description

This file defines the [OWL](#) Exception handler class and a default exception handler, for which a special class is created.

Version

Id

[class.exceptionhandler.php](#),v 1.5 2010-10-04 17:40:40 oscar Exp

7.13 /home/oscar/projects/owl-php/src/kernel/so/class.filehandler.php File Reference

Classes

- class [FileHandler](#)

File handler.

- class [OldOFMStuff](#)

Enumerations

Dataline trim flags

These flags define how datalines should be trimmed when read from a file

- enum [FILE_NOTRIM](#)

Don't trim.

- enum [FILE_TRIM_L](#)

Trim left part of a line read.

- enum [FILE_TRIM_R](#)

Trim right part of a line read.

- enum [FILE_TRIM_C](#)

Replace all multiple spaces in a line read with a single space.

7.13.1 Detailed Description

This file defines the [FileHandler](#) class

Version

Id

[class.filehandler.php](#),v 1.5 2010-12-03 12:07:42 oscar Exp

7.13.2 Enumeration Type Documentation

7.13.2.1 enum [FILE_NOTRIM](#)

Don't trim.

7.13.2.2 enum [FILE_TRIM_C](#)

Replace all multiple spaces in a line read with a single space.

7.13.2.3 enum FILE_TRIM_L

Trim left part of a line read.

7.13.2.4 enum FILE_TRIM_R

Trim right part of a line read.

7.14 /home/oscar/projects/owl-php/src/kernel/so/class.formhandler.php File Reference

Classes

- class [FormHandler](#)
Formhandler.

Enumerations

Format flags

These flags define how formdata should be returned

- enum [FORMDATA_STRICT](#)
Remove all HTML tags before returning anything.
- enum [FORMDATA_BASIC_HTML](#)
Remove all tags except bold, underline etc, lists, line breaks, headers and such.
- enum [FORMDATA_EXTENDED_HTML](#)
Remove all tags except the basic HTML tags, font and table tags, divs, spans, images and anchors.
- enum [FORMDATA_FULL_HTML](#)
Remove all tags except extended HTML tags, frame, style and form (input) tags.
- enum [FORMDATA_CUSTOM](#)
Provide custom arrays for formatting,.
- enum [FORMDATA_HTML_CODE](#)
Return all data, formatted as HTML code.
- enum [FORMDATA_RAW](#)
Return the data unformatted.

7.14.1 Detailed Description

This file defines the Formhandler class

Version

Id

[class.formhandler.php](#), v 1.6 2010-12-03 12:07:42 oscar Exp

7.14.2 Enumeration Type Documentation

7.14.2.1 enum [FORMDATA_BASIC_HTML](#)

Remove all tags except bold, underline etc, lists, line breaks, headers and such.

7.14.2.2 enum FORMDATA_CUSTOM

Provide custom arrays for formatting,.

See also

`cleanString()`

7.14.2.3 enum FORMDATA_EXTENDED_HTML

Remove all tags except the basic HTML tags, font and table tags, divs, spans, images and anchors.

7.14.2.4 enum FORMDATA_FULL_HTML

Remove all tags except extended HTML tags, frame, style and form (input) tags.

7.14.2.5 enum FORMDATA_HTML_CODE

Return all data, formatted as HTML code.

7.14.2.6 enum FORMDATA_RAW

Return the data unformatted.

7.14.2.7 enum FORMDATA_STRICT

Remove all HTML tags before returning anything.

7.15 /home/oscar/projects/owl-php/src/kernel/so/class.imagehandler.php File Reference

Classes

- class [ImageHandler](#)
Image handler.

Enumerations

Dataline trim flags

These flags define how datalines should be trimmed when read from a file

- enum [FILE_NOTRIM](#)
Don't trim.
- enum [FILE_TRIM_L](#)
Trim left part of a line read.
- enum [FILE_TRIM_R](#)
Trim right part of a line read.
- enum [FILE_TRIM_C](#)
Replace all multiple spaces in a line read with a single space.

7.15.1 Detailed Description

This file defines the [FileHandler](#) class

Version

Id

[class.imagehandler.php](#), v 1.1 2010-12-03 12:07:42 oscar Exp

7.15.2 Enumeration Type Documentation

7.15.2.1 enum FILE_NOTRIM

Don't trim.

7.15.2.2 enum FILE_TRIM_C

Replace all multiple spaces in a line read with a single space.

7.15.2.3 enum FILE_TRIM_L

Trim left part of a line read.

7.15.2.4 enum FILE_TRIM_R

Trim right part of a line read.

7.16 /home/oscar/projects/owl-php/src/kernel/so/class.loghandler.php File Reference

Classes

- class [LogHandler](#)

Log handler.

7.16.1 Detailed Description

This file defines the Loghandler class

Version

Id

[class.loghandler.php](#),v 1.5 2010-10-04 17:40:40 oscar Exp

7.17 /home/oscar/projects/owl-php/src/kernel/so/class.register.php File Reference

Classes

- class [Register](#)

Enumerations

Bitmaps

These bitmaps define the layout of status codes. They are used to extract information from the code

- enum [OWL_APPLICATION_PATTERN](#)
- enum [OWL_OBJECT_PATTERN](#)
- enum [OWL_STATUS_PATTERN](#)
- enum [OWL_SEVERITY_PATTERN](#)

7.17.1 Detailed Description

Define the abstract [Register](#) class.

Version

Id

[class.register.php](#),v 1.2 2010-08-20 08:39:54 oscar Exp

7.17.2 Enumeration Type Documentation

7.17.2.1 enum OWL_APPLICATION_PATTERN

Bits 1-8 define the application. Application identifiers with the first bit set (0x40 - 0xff) are reserved for Oveas. 0xff is the [OWL](#) Identifier.

7.17.2.2 enum OWL_OBJECT_PATTERN

Bits 9-20 define the object type of an application

7.17.2.3 enum OWL_SEVERITY_PATTERN

Bits 29-32 define the severity

7.17.2.4 enum OWL_STATUS_PATTERN

Bits 21-28 defines the (object specific) status code

7.18 /home/oscar/projects/owl-php/src/kernel/so/class.schemehandler.php File Reference

Classes

- class [SchemeHandler](#)

Scheme handler.

7.18.1 Detailed Description

This file defines the Scheme Handler class

Version

Id

[class.schemehandler.php](#),v 1.2 2010-12-03 12:07:42 oscar Exp

7.19 /home/oscar/projects/owl-php/src/kernel/so/class.sessionhandler.h File Reference

Classes

- class [SessionHandler](#)
the PHP session object

Enumerations

Session variable Flags

These flags that define how to treat values when setting session variables

- enum [SESSIONVAR_SET](#)
Set variable to the given value (default).
- enum [SESSIONVAR_UNSET](#)
Unset the variable.
- enum [SESSIONVAR_INCR](#)
Increase the variable or set as the given value if not yet existing.
- enum [SESSIONVAR_DECR](#)
Decrease the variable or set as the given value if not yet existing.
- enum [SESSIONVAR_ARRAY](#)
Add the variable to an array. If a value already exists, it will be the first element.

7.19.1 Detailed Description

This file defines the [SessionHandler](#) class

Version

Id

[class.sessionhandler.php](#),v 1.6 2010-12-03 12:07:42 oscar Exp

7.19.2 Enumeration Type Documentation

7.19.2.1 enum SESSIONVAR_ARRAY

Add the variable to an array. If a value already exists, it will be the first element.

7.19.2.2 enum SESSIONVAR_DECR

Decrease the variable or set as the given value if not yet existing.

7.19.2.3 enum SESSIONVAR_INCR

Increase the variable or set as the given value if not yet existing.

7.19.2.4 enum SESSIONVAR_SET

Set variable to the given value (default).

7.19.2.5 enum SESSIONVAR_UNSET

Unset the variable.

7.20 /home/oscar/projects/owl-php/src/kernel/so/class.statushandler.p File Reference

Classes

- class [StatusHandler](#)

Status object.

7.20.1 Detailed Description

This file defines status object that's user for all objects

Version

Id

[class.statushandler.php](#),v 1.4 2010-10-04 17:40:40 oscar Exp

7.21 /home/oscar/projects/owl-php/src/kernel/so/class.userhandler.php File Reference

Classes

- class [UserHandler](#)

the user object

7.21.1 Detailed Description

This file defines the [UserHandler](#) class

Version

Id

[class.userhandler.php,v](#) 1.9 2010-12-03 12:07:42 oscar Exp

7.22 /home/oscar/projects/owl-php/src/kernel/ui/class.baseelement.php File Reference

Classes

- class [BaseElement](#)
DOM Element base class.

7.22.1 Detailed Description

This file defines the top-level [BaseElement](#) class

Version

Id

[class.baseelement.php](#),v 1.1 2010-12-03 12:07:43 oscar Exp

7.23 /home/oscar/projects/owl-php/src/kernel/ui/class.formfield.php File Reference

Classes

- class [FormField](#)

Formfield.

7.23.1 Detailed Description

This file defines a formfield element

Version

Id

[class.formfield.php](#),v 1.1 2010-12-03 12:07:43 oscar Exp

7.24 /home/oscar/projects/owl-php/src/kernel/ui/formfields/class.form File Reference

Classes

- class [FormFieldButton](#)

Formfield.

7.24.1 Detailed Description

This file defines a button formfield element

Version

Id

[class.formfield.button.php](#),v 1.1 2010-12-03 12:07:42 oscar Exp

7.25 /home/oscar/projects/owl-
php/src/kernel/ui/formfields/class.formfield.checkbox.php File

Reference [7.25](#) /home/oscar/projects/owl-php/src/kernel/ui/formfields/class.form
File Reference ²⁷¹

Classes

- class [FormFieldCheckbox](#)

Formfield.

7.25.1 Detailed Description

This file defines a checkbox formfield element

Version

Id

[class.formfield.checkbox.php](#), v 1.1 2010-12-03 12:07:42 oscar Exp

7.26 /home/oscar/projects/owl-php/src/kernel/ui/formfields/class.form File Reference

Classes

- class [FormFieldFile](#)

Formfield.

7.26.1 Detailed Description

This file defines a fileinput formfield element

Version

Id

[class.formfield.file.php](#),v 1.1 2010-12-03 12:07:42 oscar Exp

7.27

/home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.radio.php

File Reference

7.27 /home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.radio.php
File Reference ²⁷³

Classes

- class [FormFieldRadio](#)

Formfield.

7.27.1 Detailed Description

This file defines a radio formfield element

Version

Id

[class.formfield.radio.php](#),v 1.1 2010-12-03 12:07:42 oscar Exp

7.28 /home/oscar/projects/owl-php/src/kernel/ui/formfields/class.form File Reference

Classes

- class [FormFieldSelect](#)

Formfield.

7.28.1 Detailed Description

This file defines a selectlist formfield element

Version

Id

[class.formfield.select.php](#),v 1.1 2010-12-03 12:07:42 oscar Exp

7.29

/home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.text.php

File Reference

7.29 /home/oscar/projects/owl-php/src/kernel/ui/formfields/class.formfield.text.php
File Reference 275

Classes

- class [FormFieldText](#)

Formfield.

7.29.1 Detailed Description

This file defines a text-, password or hidden formfield element

Version

Id

[class.formfield.text.php](#),v 1.1 2010-12-03 12:07:42 oscar Exp

7.30 /home/oscar/projects/owl-php/src/kernel/ui/formfields/class.form File Reference

Classes

- class [FormFieldTextarea](#)

Formfield.

7.30.1 Detailed Description

This file defines a textarea formfield element

Version

Id

[class.formfield.textarea.php,v](#) 1.1 2010-12-03 12:07:42 oscar Exp

7.31 /home/oscar/projects/owl-php/src/lib/owl.debug.functions.php File Reference

Functions

- [DBG_dumpval \(&\\$var\)](#)

7.31.1 Detailed Description

This file defines helper functions in debug mode

Version

Id

[owl.debug.functions.php](#), v 1.1 2010-12-03 12:07:42 oscar Exp

7.31.2 Function Documentation

7.31.2.1 [DBG_dumpval \(&\\$ var\)](#)

7.32 /home/oscar/projects/owl-php/src/lib/owl.helper.functions.php File Reference

7.32.1 Detailed Description

This file defines general helper functions

Version

Id

[owl.helper.functions.php](#),v 1.1 2010-12-03 12:07:42 oscar Exp

7.33 /home/oscar/projects/owl-php/src/lib/owl.messages.en-uk.php File Reference

7.33 /home/oscar/projects/owl-php/src/lib/owl.messages.en-uk.php File Reference

Variables

- **\$GLOBALS** ['messages']

7.33.1 Detailed Description

This file defines the message text for all status codes in UK English

Version

Id

owl.messages.en-uk.php,v 1.9 2010-10-04 17:40:40 oscar Exp

7.33.2 Variable Documentation

7.33.2.1 \$GLOBALS['messages']

7.34 /home/oscar/projects/owl-php/src/lib/owl.messages.php File Reference

Variables

- **\$GLOBALS** ['messages']

7.34.1 Detailed Description

This file defines the message text for all status codes in UK English

Version

Id

[owl.messages.php](#),v 1.10 2010-12-03 12:07:42 oscar Exp

7.34.2 Variable Documentation

7.34.2.1 \$GLOBALS['messages']

7.35 /home/oscar/projects/owl-php/src/lib/owl.nodebug.functions.php File Reference

Functions

- [DBG_dumpval](#) (&\$var)

7.35.1 Detailed Description

This file defines helper functions when not in debug mode

Version

Id

[owl.nodebug.functions.php](#),v 1.1 2010-12-03 12:07:42 oscar Exp

7.35.2 Function Documentation

7.35.2.1 [DBG_dumpval](#) (&\$ var)

7.36 /home/oscar/projects/owl-php/src/lib/owl.severitycodes.php File Reference

Enumerations

Severity codes

Define the Severity values for all status codes as returned by `_OWL::severity()`.

- enum `OWL_DEBUG`
- enum `OWL_INFO`
- enum `OWL_OK`
- enum `OWL_SUCCESS`
- enum `OWL_WARNING`
- enum `OWL_BUG`
- enum `OWL_ERROR`
- enum `OWL_FATAL`
- enum `OWL_CRITICAL`

7.36.1 Detailed Description

This file defines the Severity codes that all objects can have. It must be the very first file that is included, since these codes are used from step 1.

Version

Id

`owl.severitycodes.php`, v 1.2 2008-08-22 12:02:13 oscar Exp

7.36.2 Enumeration Type Documentation

7.36.2.1 enum `OWL_BUG`

Something weird happened, this might be a bug

7.36.2.2 enum `OWL_CRITICAL`

Pack your backs and start running. This status is reserved

7.36.2.3 enum `OWL_DEBUG`

Status that can be logged in debug mode

7.36.2.4 enum `OWL_ERROR`

Last operational ended with an error; current object cannot be trusted anymore

7.36.2.5 enum `OWL_FATAL`

Last operation had a fatal status; `OWL` environment cannot be trusted anymore

7.36.2.6 enum OWL_INFO

Holds neutral information about the current object status

7.36.2.7 enum OWL_OK

General normal status

7.36.2.8 enum OWL_SUCCESS

Last operation ended successful

7.36.2.9 enum OWL_WARNING

Last operation ended with a warning; this might be a temporary status (e.g. network issue) or a user error

7.37 /home/oscar/projects/owl-php/src/OWLloader.php File Reference

Classes

- class [OWLloader](#)

Class loader.

Enumerations

- enum [OWL_INCLUDE](#)
- enum [OWL_SO_INC](#)
- enum [OWL_BO_INC](#)
- enum [OWL_UI_INC](#)
- enum [OWL_LIBRARY](#)
- enum [OWL_SITE_TOP](#)

Variables

- [\\$GLOBALS](#) [‘OWLCache’]
- [\\$GLOBALS](#) [‘messages’] = array ()
- [if](#) ([array_key_exists](#)(‘app’, [ConfigHandler::get](#)(‘configfiles’)))
- [\\$GLOBALS](#) [‘logger’] = [OWL::factory](#)(‘[LogHandler](#)’)

7.37.1 Detailed Description

This file loads the [OWL](#) environment and initialises some singlettons

Version

Id

[OWLloader.php](#),v 1.10 2010-12-03 12:07:43 oscar Exp

7.37.2 Enumeration Type Documentation

7.37.2.1 enum OWL_BO_INC

7.37.2.2 enum OWL_INCLUDE

7.37.2.3 enum OWL_LIBRARY

7.37.2.4 enum OWL_SITE_TOP

7.37.2.5 enum OWL_SO_INC

7.37.2.6 enum OWL_UI_INC

7.37.3 Variable Documentation

7.37.3.1 \$GLOBALS['logger'] = OWL::factory('LogHandler')

7.37.3.2 \$GLOBALS['messages'] = array ()

7.37.3.3 \$GLOBALS['OWLCache']

Initial value:

```
array (
    'classesLoaded' => array()
)
```

7.37.3.4 if(array_key_exists('app', ConfigHandler::get('configfiles')))

7.38 /home/oscar/projects/owl-php/src/OWLrundown.php File Reference

Variables

- `$_form = OWL::factory('FormHandler')`

7.38.1 Detailed Description

Make sure all objects are destroyed in the proper order

Version

Id

[OWLrundown.php](#),v 1.4 2010-10-04 17:40:40 oscar Exp

7.38.2 Variable Documentation

7.38.2.1 `$_form = OWL::factory('FormHandler')`

Index

- /home/oscar/projects/owl-
 php/src/OWLloader.php, [284](#)
- /home/oscar/projects/owl-
 php/src/OWLRundown.php, [286](#)
- /home/oscar/projects/owl-php/src/config.php,
 [239](#)
- /home/oscar/projects/owl-php/src/index.php,
 [241](#)
- /home/oscar/projects/owl-
 php/src/kernel/bo/class.dispatcher.php, [243](#)
- /home/oscar/projects/owl-
 php/src/kernel/bo/class.form.php,
 [244](#)
- /home/oscar/projects/owl-
 php/src/kernel/bo/class.owl.php,
 [245](#)
- /home/oscar/projects/owl-
 php/src/kernel/bo/class.session.php,
 [246](#)
- /home/oscar/projects/owl-
 php/src/kernel/bo/class.user.php,
 [247](#)
- /home/oscar/projects/owl-
 php/src/kernel/class._owl.php,
 [248](#)
- /home/oscar/projects/owl-
 php/src/kernel/so/class.confighandler.php,
 [249](#)
- /home/oscar/projects/owl-
 php/src/kernel/so/class.datahandler.php,
 [250](#)
- /home/oscar/projects/owl-
 php/src/kernel/so/class.dbhandler.php,
 [252](#)
- /home/oscar/projects/owl-
 php/src/kernel/so/class.exceptionhandler.php,
 [254](#)
- /home/oscar/projects/owl-
 php/src/kernel/so/class.filehandler.php,
 [255](#)
- /home/oscar/projects/owl-
 php/src/kernel/so/class.formhandler.php,
 [257](#)
- /home/oscar/projects/owl-
 php/src/kernel/so/class.imagehandler.php,
 [259](#)
- /home/oscar/projects/owl-
 php/src/kernel/so/class.loghandler.php,
 [261](#)
- /home/oscar/projects/owl-
 php/src/kernel/so/class.register.php,
 [262](#)
- /home/oscar/projects/owl-
 php/src/kernel/so/class.schemehandler.php,
 [263](#)
- /home/oscar/projects/owl-
 php/src/kernel/so/class.sessionhandler.php,
 [264](#)
- /home/oscar/projects/owl-
 php/src/kernel/so/class.statushandler.php,
 [266](#)
- /home/oscar/projects/owl-
 php/src/kernel/so/class.userhandler.php,
 [267](#)
- /home/oscar/projects/owl-
 php/src/kernel/ui/class.baseelement.php,
 [268](#)
- /home/oscar/projects/owl-
 php/src/kernel/ui/class.formfield.php,
 [269](#)
- /home/oscar/projects/owl-
 php/src/kernel/ui/formfields/class.formfield.button.php,
 [270](#)
- /home/oscar/projects/owl-
 php/src/kernel/ui/formfields/class.formfield.checkbox.php,
 [271](#)
- /home/oscar/projects/owl-
 php/src/kernel/ui/formfields/class.formfield.file.php,
 [272](#)
- /home/oscar/projects/owl-
 php/src/kernel/ui/formfields/class.formfield.radio.php,
 [273](#)
- /home/oscar/projects/owl-
 php/src/kernel/ui/formfields/class.formfield.select.php,
 [274](#)
- /home/oscar/projects/owl-
 php/src/kernel/ui/formfields/class.formfield.text.php,
 [275](#)

/home/oscar/projects/owl-
 php/src/kernel/ui/formfields/class.formfield.**FormFieldSelect**, 129
 276
 /home/oscar/projects/owl-
 php/src/lib/owl.debug.functions.php,
 277
 /home/oscar/projects/owl-
 php/src/lib/owl.helper.functions.php,
 278
 /home/oscar/projects/owl-
 php/src/lib/owl.messages.en-uk.php,
 279
 /home/oscar/projects/owl-
 php/src/lib/owl.messages.php, **280**
 /home/oscar/projects/owl-
 php/src/lib/owl.nodebug.functions.php,
 281
 /home/oscar/projects/owl-
 php/src/lib/owl.severitycodes.php,
 282

\$GLOBALS
 config.php, **239**
 owl.messages.en-uk.php, **279**
 owl.messages.php, **280**
 OWLloader.php, **285**

\$LoginForm
 index.php, **242**

\$_form
 index.php, **241**
 OWLRundown.php, **286**

\$_index
 index.php, **241**

\$_scheme
 index.php, **242**

\$_t
 index.php, **242**

\$_table
 index.php, **242**

\$_user
 index.php, **242**

\$salt
 FormFieldButton, **92**

\$caller
 OWLException, **184**

\$checked
 FormFieldCheckbox, **101**

\$class
 BaseElement, **24**
 Form, **74**
 FormField, **83**
 FormFieldButton, **92**
 FormFieldCheckbox, **101**
 FormFieldFile, **110**
 FormFieldRadio, **119**

\$code
 StatusHandler, **221**

\$cols
 FormFieldTextarea, **148**

\$database
 DbHandler, **50**

\$dataset
 Session, **210**
 SessionHandler, **218**
 User, **230**
 UserHandler, **238**

\$db
 SchemeHandler, **201**
 SessionHandler, **218**

\$db_prefix
 DbHandler, **50**

\$disabled
 FormField, **83**
 FormFieldButton, **92**
 FormFieldCheckbox, **101**
 FormFieldFile, **110**
 FormFieldRadio, **119**
 FormFieldSelect, **129**
 FormFieldText, **139**
 FormFieldTextarea, **148**

\$dispatcher
 Form, **74**

\$errno
 DbHandler, **51**

\$error
 DbHandler, **51**

\$fields
 Form, **74**

\$filename
 LogHandler, **173**

\$fpointer
 FileHandler, **65**
 ImageHandler, **164**
 LogHandler, **173**

\$hidden_args
 OWLException, **184**

\$id
 BaseElement, **24**
 DbHandler, **51**
 Form, **74**
 FormField, **83**
 FormFieldButton, **92**
 FormFieldCheckbox, **101**
 FormFieldFile, **110**
 FormFieldRadio, **119**
 FormFieldSelect, **129**

FormFieldText, 139
FormFieldTextarea, 148
SchemeHandler, 201
\$instance
 DbHandler, 51
 Dispatcher, 58
 FormHandler, 157
 LogHandler, 173
 OWL, 181
 SchemeHandler, 201
 StatusHandler, 221
\$inuse
 SchemeHandler, 201
\$maxsize
 FormFieldText, 139
\$multiple
 FormFieldSelect, 129
\$name
 FileHandler, 65
 FormField, 83
 FormFieldButton, 92
 FormFieldCheckbox, 102
 FormFieldFile, 110
 FormFieldRadio, 119
 FormFieldSelect, 129
 FormFieldText, 139
 FormFieldTextarea, 148
 ImageHandler, 164
\$opened
 DbHandler, 51
 FileHandler, 66
 ImageHandler, 165
 LogHandler, 173
\$options
 FormFieldSelect, 129
\$original_name
 FileHandler, 66
 ImageHandler, 165
\$owl_data
 DataHandler, 36
\$owl_database
 DataHandler, 36
\$owl_formvalues
 FormHandler, 157
\$owl_joins
 DataHandler, 36
\$owl_keys
 DataHandler, 36
\$owl_prepared
 DataHandler, 36
\$owl_tablename
 DataHandler, 36
\$params
 StatusHandler, 221
\$pstatus
 OWL, 17
\$query
 DbHandler, 51
\$readonly
 FormField, 83
 FormFieldButton, 92
 FormFieldCheckbox, 102
 FormFieldFile, 110
 FormFieldRadio, 120
 FormFieldSelect, 129
 FormFieldText, 139
 FormFieldTextarea, 148
\$rowcount
 DbHandler, 51
\$rows
 FormFieldTextarea, 149
\$scheme
 SchemeHandler, 201
\$selected
 FormFieldRadio, 120
\$session
 User, 230
 UserHandler, 238
\$severity
 OWL, 17
 BaseElement, 24
 DataHandler, 37
 DbHandler, 51
 Dispatcher, 58
 FileHandler, 66
 Form, 74
 FormField, 83
 FormFieldButton, 93
 FormFieldCheckbox, 102
 FormFieldFile, 110
 FormFieldRadio, 120
 FormFieldSelect, 129
 FormFieldText, 139
 FormFieldTextarea, 149
 FormHandler, 157
 ImageHandler, 165
 LogHandler, 173
 OWL, 181
 SchemeHandler, 202
 Session, 210
 SessionHandler, 218
 User, 230
 UserHandler, 238
\$size
 FormFieldSelect, 129
 FormFieldText, 139
\$src
 FormFieldButton, 93

\$status
 — _OWL, 17
\$style
 BaseElement, 24
 Form, 75
 FormField, 83
 FormFieldButton, 93
 FormFieldCheckbox, 102
 FormFieldFile, 110
 FormFieldRadio, 120
 FormFieldSelect, 130
 FormFieldText, 139
 FormFieldTextarea, 149
\$table
 SchemeHandler, 202
\$thrown_code
 OWLException, 184
\$type
 FormField, 83
 FormFieldButton, 93
 FormFieldCheckbox, 102
 FormFieldFile, 110
 FormFieldRadio, 120
 FormFieldSelect, 130
 FormFieldText, 140
 FormFieldTextarea, 149
\$user_data
 User, 230
 UserHandler, 238
\$value
 FormField, 84
 FormFieldButton, 93
 FormFieldCheckbox, 102
 FormFieldFile, 111
 FormFieldRadio, 120
 FormFieldSelect, 130
 FormFieldText, 140
 FormFieldTextarea, 149
_OWL, 13
 \$pstatus, 17
 \$severity, 17
 \$status, 17
 — — destruct, 14
 check, 15
 get_severity, 15
 get_status, 15
 init, 15
 reset, 16
 reset_calltree, 16
 set_high_severity, 16
 set_status, 16
 signal, 17
 traceback, 17
 — clone
 — —

DbHandler, 41
 Dispatcher, 54
 LogHandler, 169
 OWL, 177
 SchemeHandler, 195
 StatusHandler, 220
 — construct
 DbHandler, 41
 Dispatcher, 54
 FileHandler, 61
 Form, 69
 FormField, 78
 FormFieldButton, 87
 FormFieldCheckbox, 96
 FormFieldFile, 105
 FormFieldRadio, 114
 FormFieldSelect, 124
 FormFieldText, 133
 FormFieldTextarea, 143
 FormHandler, 152
 ImageHandler, 160
 LogHandler, 168
 OWL, 177
 OWLException, 182
 SchemeHandler, 195
 Session, 205
 SessionHandler, 213
 StatusHandler, 220
 User, 225
 UserHandler, 233
 — destruct
 — _OWL, 14
 DbHandler, 41
 FileHandler, 61
 LogHandler, 168
 SchemeHandler, 195
 Session, 205
 SessionHandler, 213
 User, 225
 UserHandler, 233
 — define_field
 SchemeHandler, 195
 — tryLoad
 OWLloader, 186

addField
 Form, 70
 alter_scheme
 SchemeHandler, 195
 alter_table
 SchemeHandler, 195

backtrace
 LogHandler, 169

BaseElement, 19
 \$class, 24
 \$id, 24
 \$severity, 24
 \$style, 24
 check, 21
 get_severity, 21
 get_status, 22
 getAttributes, 22
 getEvents, 22
 init, 22
 reset, 22
 set_high_severity, 22
 set_status, 23
 setClass, 23
 setId, 23
 signal, 23
 traceback, 24

Business Object modules, 10

check
 OWL, 15
 BaseElement, 21
 DataHandler, 30
 DbHandler, 41
 Dispatcher, 54
 FileHandler, 62
 Form, 70
 FormField, 78
 FormFieldButton, 88
 FormFieldCheckbox, 97
 FormFieldFile, 105
 FormFieldRadio, 115
 FormFieldSelect, 124
 FormFieldText, 134
 FormFieldTextarea, 144
 FormHandler, 153
 ImageHandler, 161
 LogHandler, 169
 OWL, 177
 SchemeHandler, 196
 Session, 205
 SessionHandler, 214
 User, 226
 UserHandler, 234

check_hide
 OWLEception, 183

class.datahandler.php
 DATA_DELETE, 251
 DATA_READ, 251
 DATA_RESET_FULL, 251
 DATA_RESET_META, 251
 DATA_RESET_PREPARE, 251
 DATA_RESET_STATUS, 251

DATA_UNPREPARED, 251
DATA_UPDATE, 251
DATA_WRITE, 251

class.dbhandler.php
 DBHANDLE_DATA, 252
 DBHANDLE_FIELDCOUNT, 252
 DBHANDLE_ROWCOUNT, 253
 DBHANDLE_SINGLEFIELD, 253
 DBHANDLE_SINGLEROW, 253
 DBHANDLE_TOTALFIELDCOUNT, 253

class.filehandler.php
 FILE_NOTRIM, 255
 FILE_TRIM_C, 255
 FILE_TRIM_L, 255
 FILE_TRIM_R, 256

class.formhandler.php
 FORMDATA_BASIC_HTML, 257
 FORMDATA_CUSTOM, 257
 FORMDATA_EXTENDED_HTML, 258
 FORMDATA_FULL_HTML, 258
 FORMDATA_HTML_CODE, 258
 FORMDATA_RAW, 258
 FORMDATA_STRICT, 258

class.imagehandler.php
 FILE_NOTRIM, 259
 FILE_TRIM_C, 259
 FILE_TRIM_L, 259
 FILE_TRIM_R, 259

class.register.php
 OWL_APPLICATION_PATTERN, 262
 OWL_OBJECT_PATTERN, 262
 OWL_SEVERITY_PATTERN, 262
 OWL_STATUS_PATTERN, 262

class.sessionhandler.php
 SESSIONVAR_ARRAY, 264
 SESSIONVAR_DECR, 264
 SESSIONVAR_INCR, 264
 SESSIONVAR_SET, 265
 SESSIONVAR_UNSET, 265

cleanString
 FormHandler, 153

close
 DbHandler, 42
 FileHandler, 62
 ImageHandler, 161
 Session, 206
 SessionHandler, 214

close_logfile
 LogHandler, 169

closeForm
 Form, 70

compare
 SchemeHandler, 196

compose_message
 LogHandler, 169
 config.php
 \$GLOBALS, 239
 ConfigHandler, 26
 convert, 26
 get, 26
 read_config, 27
 set, 27
 connect
 DbHandler, 42
 convert
 ConfigHandler, 26
 create
 DbHandler, 42
 create_scheme
 SchemeHandler, 196
 create_table
 SchemeHandler, 196

 DATA_DELETE
 class.datahandler.php, 251
 DATA_READ
 class.datahandler.php, 251
 DATA_RESET_FULL
 class.datahandler.php, 251
 DATA_RESET_META
 class.datahandler.php, 251
 DATA_RESET_PREPARE
 class.datahandler.php, 251
 DATA_RESET_STATUS
 class.datahandler.php, 251
 DATA_UNPREPARED
 class.datahandler.php, 251
 DATA_UPDATE
 class.datahandler.php, 251
 DATA_WRITE
 class.datahandler.php, 251
 DataHandler, 28
 \$owl_data, 36
 \$owl_database, 36
 \$owl_joins, 36
 \$owl_keys, 36
 \$owl_prepared, 36
 \$owl_tablename, 36
 \$severity, 37
 check, 30
 DataHandler, 31
 db, 31
 db_status, 31
 escape_string, 31
 find_field, 32
 get, 32
 get_severity, 32
 get_status, 32
 init, 33
 prepare, 33
 reset, 33
 set, 34
 set_high_severity, 34
 set_join, 34
 set_key, 34
 set_status, 35
 set_tablename, 35
 signal, 35
 traceback, 36
 db
 DataHandler, 31
 db_status
 DataHandler, 31
 DBG_DUMPVAL
 owl.debug.functions.php, 277
 owl.nodebug.functions.php, 281
 DBHANDLE_DATA
 class.dbhandler.php, 252
 DBHANDLE_FIELDCOUNT
 class.dbhandler.php, 252
 DBHANDLE_ROWCOUNT
 class.dbhandler.php, 253
 DBHANDLE_SINGLEFIELD
 class.dbhandler.php, 253
 DBHANDLE_SINGLEROW
 class.dbhandler.php, 253
 DBHANDLE_TOTALFIELDCOUNT
 class.dbhandler.php, 253
 DbHandler, 38
 \$database, 50
 \$db_prefix, 50
 \$errno, 51
 \$error, 51
 \$fd, 51
 \$instance, 51
 \$opened, 51
 \$query, 51
 \$rowcount, 51
 \$severity, 51
 __clone, 41
 __construct, 41
 __destruct, 41
 check, 41
 close, 42
 connect, 42
 create, 42
 dbread, 42
 escape_string, 43
 expand_field, 43
 extract_tablelist, 43
 get_instance, 43

get_severity, 44
get_status, 44
init, 44
is_open, 44
last_inserted_id, 44
open, 45
prepare_delete, 45
prepare_insert, 45
prepare_read, 45
prepare_update, 46
read, 46
reset, 47
set_high_severity, 47
set_query, 47
set_status, 47
signal, 48
table_exists, 48
tablelist, 48
tablename, 49
traceback, 49
unescape_string, 49
update_list, 49
where_clause, 50
write, 50

dbread
 DbHandler, 42
DefaultOptionGroup
 FormFieldSelect, 130
define_index
 SchemeHandler, 197
define_scheme
 SchemeHandler, 197
destroy
 Session, 206
 SessionHandler, 214
dispatch
 Dispatcher, 55
Dispatcher, 52
 \$instance, 58
 \$severity, 58
 __clone, 54
 __construct, 54
 check, 54
 dispatch, 55
 get_instance, 55
 get_severity, 55
 get_status, 55
 init, 56
 reset, 56
 set_high_severity, 56
 set_status, 56
 signal, 57
 traceback, 57
download
 FileHandler, 62
 ImageHandler, 161
encode
 FileHandler, 62
 ImageHandler, 161
escape_string
 DataHandler, 31
 DbHandler, 43
expand_field
 DbHandler, 43
extract_tablelist
 DbHandler, 43
factory
 OWL, 178
FILE_NOTRIM
 class.filehandler.php, 255
 class.imagehandler.php, 259
FILE_TRIM_C
 class.filehandler.php, 255
 class.imagehandler.php, 259
FILE_TRIM_L
 class.filehandler.php, 255
 class.imagehandler.php, 259
FILE_TRIM_R
 class.filehandler.php, 256
 class.imagehandler.php, 259
FileHandler, 59
 \$fp pointer, 65
 \$name, 65
 \$opened, 66
 \$original_name, 66
 \$severity, 66
 __construct, 61
 __destruct, 61
 check, 62
 close, 62
 download, 62
 encode, 62
 get_severity, 62
 get_status, 63
 init, 63
 open, 63
 read_data, 63
 read_line, 63
 reset, 64
 set_high_severity, 64
 set_status, 64
 signal, 65
 traceback, 65
find_field
 DataHandler, 32
Form, 67

\$class, 74
 \$dispatcher, 74
 \$fields, 74
 \$id, 74
 \$severity, 74
 \$style, 75
 __construct, 69
 addField, 70
 check, 70
 closeForm, 70
 get_severity, 70
 get_status, 71
 getAttributes, 71
 getEvents, 71
 init, 71
 openForm, 71
 reset, 72
 set_high_severity, 72
 set_status, 72
 setClass, 73
 setFieldAttributes, 73
 setId, 73
 showField, 73
 signal, 73
 traceback, 74
FORMDATA_BASIC_HTML
 class.formhandler.php, 257
FORMDATA_CUSTOM
 class.formhandler.php, 257
FORMDATA_EXTENDED_HTML
 class.formhandler.php, 258
FORMDATA_FULL_HTML
 class.formhandler.php, 258
FORMDATA_HTML_CODE
 class.formhandler.php, 258
FORMDATA_RAW
 class.formhandler.php, 258
FORMDATA_STRICT
 class.formhandler.php, 258
FormField, 76
 \$class, 83
 \$disabled, 83
 \$id, 83
 \$name, 83
 \$readonly, 83
 \$severity, 83
 \$style, 83
 \$type, 83
 \$value, 84
 __construct, 78
 check, 78
 get_severity, 79
 get_status, 79
 getAttributes, 79
 getEvents, 79
 getGenericFieldAttributes, 79
 getType, 80
 init, 80
 reset, 80
 set_high_severity, 80
 set_status, 81
 setClass, 81
 setDisabled, 81
 setId, 81
 setName, 82
 setReadonly, 82
 setValue, 82
 signal, 82
 traceback, 82
FormFieldButton, 85
 \$salt, 92
 \$class, 92
 \$disabled, 92
 \$id, 92
 \$name, 92
 \$readonly, 92
 \$severity, 93
 \$src, 93
 \$style, 93
 \$type, 93
 \$value, 93
 __construct, 87
 check, 88
 get_severity, 88
 get_status, 88
 getAttributes, 88
 getEvents, 88
 getFieldCode, 89
 getGenericFieldAttributes, 89
 getType, 89
 init, 89
 reset, 89
 set_high_severity, 90
 set_status, 90
 setClass, 90
 setDisabled, 90
 setId, 91
 setName, 91
 setReadonly, 91
 setValue, 91
 signal, 91
 traceback, 92
FormFieldCheckbox, 94
 \$checked, 101
 \$class, 101
 \$disabled, 101
 \$id, 101
 \$name, 102

\$readonly, 102
\$severity, 102
\$style, 102
\$type, 102
\$value, 102
 __construct, 96
check, 97
get_severity, 97
get_status, 97
getAttributes, 97
getEvents, 97
getFieldCode, 98
getGenericFieldAttributes, 98
getType, 98
init, 98
reset, 98
set_high_severity, 99
set_status, 99
setChecked, 99
setClass, 99
setDisabled, 100
setId, 100
setName, 100
setReadOnly, 100
setValue, 100
signal, 100
traceback, 101
FormFieldFile, 103
 \$class, 110
 \$disabled, 110
 \$id, 110
 \$name, 110
 \$readonly, 110
 \$severity, 110
 \$style, 110
 \$type, 110
 \$value, 111
 __construct, 105
check, 105
get_severity, 106
get_status, 106
getAttributes, 106
getEvents, 106
getGenericFieldAttributes, 106
getType, 107
init, 107
reset, 107
set_high_severity, 107
set_status, 108
setClass, 108
setDisabled, 108
setId, 108
setName, 109
setReadOnly, 109
 setValue, 109
 signal, 109
 traceback, 109
FormFieldRadio, 112
 \$class, 119
 \$disabled, 119
 \$id, 119
 \$name, 119
 \$readonly, 120
 \$selected, 120
 \$severity, 120
 \$style, 120
 \$type, 120
 \$value, 120
 __construct, 114
check, 115
get_severity, 115
get_status, 115
getAttributes, 115
getEvents, 115
getFieldCode, 116
getGenericFieldAttributes, 116
getType, 116
init, 116
reset, 116
set_high_severity, 117
set_status, 117
setClass, 117
setDisabled, 117
setId, 118
setName, 118
setReadOnly, 118
setSelected, 118
setValue, 118
signal, 118
traceback, 119
FormFieldSelect, 121
 \$class, 129
 \$disabled, 129
 \$id, 129
 \$multiple, 129
 \$name, 129
 \$options, 129
 \$readonly, 129
 \$severity, 129
 \$size, 129
 \$style, 130
 \$type, 130
 \$value, 130
 __construct, 124
check, 124
DefaultOptionGroup, 130
get_severity, 124
get_status, 124

getAttributes, 124
 getEvents, 125
 getFieldCode, 125
 getGenericFieldAttributes, 125
 getType, 125
 init, 125
 reset, 126
 set_high_severity, 126
 set_status, 126
 setClass, 127
 setDisabled, 127
 setId, 127
 setMultiple, 127
 setName, 127
 setReadonly, 127
 setSize, 127
 setValue, 128
 signal, 128
 traceback, 128
 FormFieldText, 131
 \$class, 139
 \$disabled, 139
 \$id, 139
 \$maxsize, 139
 \$name, 139
 \$readonly, 139
 \$severity, 139
 \$size, 139
 \$style, 139
 \$type, 140
 \$value, 140
 __construct, 133
 check, 134
 get_severity, 134
 get_status, 134
 getAttributes, 134
 getEvents, 135
 getFieldCode, 135
 getGenericFieldAttributes, 135
 getType, 135
 init, 135
 reset, 136
 set_high_severity, 136
 set_status, 136
 setClass, 137
 setDisabled, 137
 setId, 137
 setMaxsize, 137
 setName, 137
 setReadonly, 137
 setSize, 138
 setValue, 138
 signal, 138
 traceback, 138
 FormFieldTextarea, 141
 \$class, 148
 \$cols, 148
 \$disabled, 148
 \$id, 148
 \$name, 148
 \$readonly, 148
 \$rows, 149
 \$severity, 149
 \$style, 149
 \$type, 149
 \$value, 149
 __construct, 143
 check, 144
 get_severity, 144
 get_status, 144
 getAttributes, 144
 getEvents, 144
 getFieldCode, 145
 getGenericFieldAttributes, 145
 getType, 145
 init, 145
 reset, 145
 set_high_severity, 146
 set_status, 146
 setClass, 146
 setDisabled, 146
 setId, 147
 setName, 147
 setReadonly, 147
 setValue, 147
 signal, 147
 traceback, 148
 FormHandler, 150
 \$instance, 157
 \$owl_formvalues, 157
 \$severity, 157
 __construct, 152
 check, 153
 cleanString, 153
 get, 153
 get_instance, 154
 get_severity, 154
 get_status, 154
 init, 154
 parse_formdata, 155
 reset, 155
 set, 155
 set_high_severity, 155
 set_status, 155
 signal, 156
 traceback, 156
 gc

Session, 206
SessionHandler, 214
get
 ConfigHandler, 26
 DataHandler, 32
 FormHandler, 153
get_code
 Register, 188
 StatusHandler, 220
get_id
 User, 226
get_instance
 DbHandler, 43
 Dispatcher, 55
 FormHandler, 154
 LogHandler, 170
 OWL, 178
 SchemeHandler, 197
 StatusHandler, 220
get_message
 StatusHandler, 220
get_run_id
 Register, 188
get_session_var
 Session, 206
 User, 226
get_severity
 OWL, 15
 BaseElement, 21
 DataHandler, 32
 DbHandler, 44
 Dispatcher, 55
 FileHandler, 62
 Form, 70
 FormField, 79
 FormFieldButton, 88
 FormFieldCheckbox, 97
 FormFieldFile, 106
 FormFieldRadio, 115
 FormFieldSelect, 124
 FormFieldText, 134
 FormFieldTextarea, 144
 FormHandler, 154
 ImageHandler, 161
 LogHandler, 170
 OWL, 178
 Register, 188
 SchemeHandler, 197
 Session, 206
 SessionHandler, 214
 StatusHandler, 220
 User, 226
 UserHandler, 234
get_severity_level
 Register, 189
 get_status
 OWL, 15
 BaseElement, 22
 DataHandler, 32
 DbHandler, 44
 Dispatcher, 55
 FileHandler, 63
 Form, 71
 FormField, 79
 FormFieldButton, 88
 FormFieldCheckbox, 97
 FormFieldFile, 106
 FormFieldRadio, 115
 FormFieldSelect, 124
 FormFieldText, 134
 FormFieldTextarea, 144
 FormHandler, 154
 ImageHandler, 161
 LogHandler, 170
 OWL, 178
 SchemeHandler, 198
 Session, 207
 SessionHandler, 215
 User, 226
 UserHandler, 234
 get_table_columns
 SchemeHandler, 198
 get_table_indexes
 SchemeHandler, 198
 get_trace
 OWLError, 183
 get_username
 User, 227
 UserHandler, 234
getAttributes
 BaseElement, 22
 Form, 71
 FormField, 79
 FormFieldButton, 88
 FormFieldCheckbox, 97
 FormFieldFile, 106
 FormFieldRadio, 115
 FormFieldSelect, 124
 FormFieldText, 134
 FormFieldTextarea, 144
getClass
 OWLloader, 186
getEvents
 BaseElement, 22
 Form, 71
 FormField, 79
 FormFieldButton, 88
 FormFieldCheckbox, 97

FormFieldFile, 106
 FormFieldRadio, 115
 FormFieldSelect, 125
 FormFieldText, 135
 FormFieldTextarea, 144
 getFieldCode
 FormFieldButton, 89
 FormFieldCheckbox, 98
 FormFieldRadio, 116
 FormFieldSelect, 125
 FormFieldText, 135
 FormFieldTextarea, 145
 getGenericFieldAttributes
 FormField, 79
 FormFieldButton, 89
 FormFieldCheckbox, 98
 FormFieldFile, 106
 FormFieldRadio, 116
 FormFieldSelect, 125
 FormFieldText, 135
 FormFieldTextarea, 145
 getType
 FormField, 80
 FormFieldButton, 89
 FormFieldCheckbox, 98
 FormFieldFile, 107
 FormFieldRadio, 116
 FormFieldSelect, 125
 FormFieldText, 135
 FormFieldTextarea, 145
 handle_exception
 OWLExceptionHandler, 185
 hash_password
 UserHandler, 235
 if
 OWLloader.php, 285
 ImageHandler, 158
 \$fp pointer, 164
 \$name, 164
 \$opened, 165
 \$original_name, 165
 \$severity, 165
 __construct, 160
 check, 161
 close, 161
 download, 161
 encode, 161
 get_severity, 161
 get_status, 161
 init, 162
 open, 162
 read_data, 162
 read_line, 162
 reset, 163
 set_high_severity, 163
 set_status, 163
 signal, 164
 traceback, 164
 index.php
 \$LoginForm, 242
 \$_form, 241
 \$_index, 241
 \$_scheme, 242
 \$_t, 242
 \$_table, 242
 \$_user, 242
 OWL_ROOT, 241
 switch, 242
 init
 OWL, 15
 BaseElement, 22
 DataHandler, 33
 DbHandler, 44
 Dispatcher, 56
 FileHandler, 63
 Form, 71
 FormField, 80
 FormFieldButton, 89
 FormFieldCheckbox, 98
 FormFieldFile, 107
 FormFieldRadio, 116
 FormFieldSelect, 125
 FormFieldText, 135
 FormFieldTextarea, 145
 FormHandler, 154
 ImageHandler, 162
 LogHandler, 170
 OWL, 179
 Register, 189
 SchemeHandler, 198
 Session, 207
 SessionHandler, 215
 User, 227
 UserHandler, 235
 is_open
 DbHandler, 44
 last_inserted_id
 DbHandler, 44
 Library (codes, messages files etc.), 12
 log
 LogHandler, 171
 log_exception
 OWLExceptionHandler, 185
 LogHandler, 166
 \$filename, 173

\$fpointer, 173
\$instance, 173
\$opened, 173
\$severity, 173
-- clone, 169
-- construct, 168
-- destruct, 168
backtrace, 169
check, 169
close_logfile, 169
compose_message, 169
get_instance, 170
get_severity, 170
get_status, 170
init, 170
log, 171
open_logfile, 171
reset, 171
set_filename, 171
set_high_severity, 171
set_status, 172
signal, 172
traceback, 172
write_logfile, 173
login
 User, 227
 UserHandler, 235
logout
 User, 227
 UserHandler, 235

OFM_FileDetails
 OldOFMStuff, 174
OFM_FileInfo
 OldOFMStuff, 174
OFM_IsAscii
 OldOFMStuff, 174
OldOFMStuff, 174
 OFM_FileDetails, 174
 OFM_FileInfo, 174
 OFM_IsAscii, 174
open
 DbHandler, 45
 FileHandler, 63
 ImageHandler, 162
 Session, 207
 SessionHandler, 215
open_logfile
 LogHandler, 171
openForm
 Form, 71
OWL, 175
 \$instance, 181
 \$severity, 181
 -- clone, 177
 -- construct, 177
 check, 177
 factory, 178
 get_instance, 178
 get_severity, 178
 get_status, 178
 init, 179
 reset, 179
 set_high_severity, 179
 set_status, 179
 signal, 180
 stat, 180
 traceback, 180
owl.debug.functions.php
 DBG_dumpval, 277
owl.messages.en-uk.php
 \$GLOBALS, 279
owl.messages.php
 \$GLOBALS, 280
owl.nodebug.functions.php
 DBG_dumpval, 281
owl.severitycodes.php
 OWL_BUG, 282
 OWL_CRITICAL, 282
 OWL_DEBUG, 282
 OWL_ERROR, 282
 OWL_FATAL, 282
 OWL_INFO, 282
 OWL_OK, 283
 OWL_SUCCESS, 283
 OWL_WARNING, 283
OWL_APPLICATION_PATTERN
 class.register.php, 262
OWL_BO_INC
 OWLloader.php, 285
OWL_BUG
 owl.severitycodes.php, 282
OWL_CRITICAL
 owl.severitycodes.php, 282
OWL_DEBUG
 owl.severitycodes.php, 282
OWL_ERROR
 owl.severitycodes.php, 282
OWL_FATAL
 owl.severitycodes.php, 282
OWL_INCLUDE
 OWLloader.php, 285
OWL_INFO
 owl.severitycodes.php, 282
OWL_LIBRARY
 OWLloader.php, 285
OWL_OBJECT_PATTERN
 class.register.php, 262

OWL_OK
 owl.severitycodes.php, 283

OWL_ROOT
 index.php, 241

OWL_SEVERITY_PATTERN
 class.register.php, 262

OWL_SITE_TOP
 OWLloader.php, 285

OWL_SO_INC
 OWLloader.php, 285

OWL_STATUS_PATTERN
 class.register.php, 262

OWL_SUCCESS
 owl.severitycodes.php, 283

OWL_UI_INC
 OWLloader.php, 285

OWL_WARNING
 owl.severitycodes.php, 283

OWLEception, 182
 \$caller, 184
 \$hidden_args, 184
 \$thrown_code, 184
 __construct, 182
 check_hide, 183
 get_trace, 183
 stack_dump, 183
 traceback, 183

OWLEceptionHandler, 185
 handle_exception, 185
 log_exception, 185

OWLloader, 186
 _tryLoad, 186
 getClass, 186

OWLloader.php
 \$GLOBALS, 285
 if, 285
 OWL_BO_INC, 285
 OWL_INCLUDE, 285
 OWL_LIBRARY, 285
 OWL_SITE_TOP, 285
 OWL_SO_INC, 285
 OWL_UI_INC, 285

OWLrundown.php
 \$_form, 286

parse_formdata
 FormHandler, 155

prepare
 DataHandler, 33

prepare_delete
 DbHandler, 45

prepare_insert
 DbHandler, 45

prepare_read

DbHandler, 45
 prepare_update
 DbHandler, 46

Presentation modules, 9

read
 DbHandler, 46
 Session, 207
 SessionHandler, 215

read_config
 ConfigHandler, 27

read_data
 FileHandler, 63
 ImageHandler, 162

read_line
 FileHandler, 63
 ImageHandler, 162

read_userdata
 UserHandler, 236

Register, 188
 get_code, 188
 get_run_id, 188
 get_severity, 188
 get_severity_level, 189
 init, 189
 register_app, 189
 register_class, 189
 register_code, 189
 register_messages, 190
 register_severity, 190
 set_application, 190
 set_class, 190
 set_severity, 190

register_app
 Register, 189

register_class
 Register, 189

register_code
 Register, 189

register_messages
 Register, 190

register_severity
 Register, 190

reset
 OWL, 16
 BaseElement, 22
 DataHandler, 33
 DbHandler, 47
 Dispatcher, 56
 FileHandler, 64
 Form, 72
 FormField, 80
 FormFieldButton, 89
 FormFieldCheckbox, 98

FormFieldFile, 107
FormFieldRadio, 116
FormFieldSelect, 126
FormFieldText, 136
FormFieldTextarea, 145
FormHandler, 155
ImageHandler, 163
LogHandler, 171
OWL, 179
SchemeHandler, 199
Session, 208
SessionHandler, 216
StatusHandler, 221
User, 227
UserHandler, 236
reset_calltree
 OWL, 16

scheme
 SchemeHandler, 199
SchemeHandler, 192
 \$db, 201
 \$id, 201
 \$instance, 201
 \$inuse, 201
 \$scheme, 201
 \$severity, 202
 \$table, 202
 __clone, 195
 __construct, 195
 __destruct, 195
 __define_field, 195
 alter_scheme, 195
 alter_table, 195
 check, 196
 compare, 196
 create_scheme, 196
 create_table, 196
 define_index, 197
 define_scheme, 197
 get_instance, 197
 get_severity, 197
 get_status, 198
 get_table_columns, 198
 get_table_indexes, 198
 init, 198
 reset, 199
 scheme, 199
 set_high_severity, 199
 set_status, 199
 signal, 200
 table_description, 200
 traceback, 200
 validate_scheme, 201

Session, 203
 \$dataset, 210
 \$severity, 210
 __construct, 205
 __destruct, 205
 check, 205
 close, 206
 destroy, 206
 gc, 206
 get_session_var, 206
 get_severity, 206
 get_status, 207
 init, 207
 open, 207
 read, 207
 reset, 208
 set_high_severity, 208
 set_session_var, 208
 set_status, 208
 signal, 209
 traceback, 209
 write, 209

SessionHandler, 211
 \$dataset, 218
 \$db, 218
 \$severity, 218
 __construct, 213
 __destruct, 213
 check, 214
 close, 214
 destroy, 214
 gc, 214
 get_severity, 214
 get_status, 215
 init, 215
 open, 215
 read, 215
 reset, 216
 set_high_severity, 216
 set_status, 216
 signal, 216
 traceback, 217
 write, 217

SESSIONVAR_ARRAY
 class.sessionhandler.php, 264

SESSIONVAR_DECR
 class.sessionhandler.php, 264

SESSIONVAR_INCR
 class.sessionhandler.php, 264

SESSIONVAR_SET
 class.sessionhandler.php, 265

SESSIONVAR_UNSET
 class.sessionhandler.php, 265

set

ConfigHandler, 27
 DataHandler, 34
 FormHandler, 155
 set_application
 Register, 190
 set_class
 Register, 190
 set_code
 StatusHandler, 221
 set_filename
 LogHandler, 171
 set_high_severity
 OWL, 16
 BaseElement, 22
 DataHandler, 34
 DbHandler, 47
 Dispatcher, 56
 FileHandler, 64
 Form, 72
 FormField, 80
 FormFieldButton, 90
 FormFieldCheckbox, 99
 FormFieldFile, 107
 FormFieldRadio, 117
 FormFieldSelect, 126
 FormFieldText, 136
 FormFieldTextarea, 146
 FormHandler, 155
 ImageHandler, 163
 LogHandler, 171
 OWL, 179
 SchemeHandler, 199
 Session, 208
 SessionHandler, 216
 User, 228
 UserHandler, 236
 set_join
 DataHandler, 34
 set_key
 DataHandler, 34
 set_params
 StatusHandler, 221
 set_query
 DbHandler, 47
 set_session_var
 Session, 208
 User, 228
 set_severity
 Register, 190
 set_status
 OWL, 16
 BaseElement, 23
 DataHandler, 35
 DbHandler, 47
 Dispatcher, 56
 FileHandler, 64
 Form, 72
 FormField, 81
 FormFieldButton, 90
 FormFieldCheckbox, 99
 FormFieldFile, 108
 FormFieldRadio, 117
 FormFieldSelect, 126
 FormFieldText, 136
 FormFieldTextarea, 146
 FormHandler, 155
 ImageHandler, 163
 LogHandler, 172
 OWL, 179
 SchemeHandler, 199
 Session, 208
 SessionHandler, 216
 User, 228
 UserHandler, 236
 set tablename
 DataHandler, 35
 set_username
 User, 229
 UserHandler, 237
 setChecked
 FormFieldCheckbox, 99
 setClass
 BaseElement, 23
 Form, 73
 FormField, 81
 FormFieldButton, 90
 FormFieldCheckbox, 99
 FormFieldFile, 108
 FormFieldRadio, 117
 FormFieldSelect, 127
 FormFieldText, 137
 FormFieldTextarea, 146
 setDisabled
 FormField, 81
 FormFieldButton, 90
 FormFieldCheckbox, 100
 FormFieldFile, 108
 FormFieldRadio, 117
 FormFieldSelect, 127
 FormFieldText, 137
 FormFieldTextarea, 146
 setFieldAttributes
 Form, 73
 setId
 BaseElement, 23
 Form, 73
 FormField, 81
 FormFieldButton, 91

FormFieldCheckbox, 100
FormFieldFile, 108
FormFieldRadio, 118
FormFieldSelect, 127
FormFieldText, 137
FormFieldTextarea, 147
setMaxsize
 FormFieldText, 137
setMultiple
 FormFieldSelect, 127
setName
 FormField, 82
 FormFieldButton, 91
 FormFieldCheckbox, 100
 FormFieldFile, 109
 FormFieldRadio, 118
 FormFieldSelect, 127
 FormFieldText, 137
 FormFieldTextarea, 147
set Readonly
 FormField, 82
 FormFieldButton, 91
 FormFieldCheckbox, 100
 FormFieldFile, 109
 FormFieldRadio, 118
 FormFieldSelect, 127
 FormFieldText, 137
 FormFieldTextarea, 147
set Selected
 FormFieldRadio, 118
setSize
 FormFieldSelect, 127
 FormFieldText, 138
setValue
 FormField, 82
 FormFieldButton, 91
 FormFieldCheckbox, 100
 FormFieldFile, 109
 FormFieldRadio, 118
 FormFieldSelect, 128
 FormFieldText, 138
 FormFieldTextarea, 147
showField
 Form, 73
signal
 OWL, 17
 BaseElement, 23
 DataHandler, 35
 DbHandler, 48
 Dispatcher, 57
 FileHandler, 65
 Form, 73
 FormField, 82
 FormFieldButton, 91
 FormFieldCheckbox, 100
 FormFieldFile, 109
 FormFieldRadio, 118
 FormFieldSelect, 128
 FormFieldText, 138
 FormFieldTextarea, 147
 FormHandler, 156
 ImageHandler, 164
 LogHandler, 172
 OWL, 180
 SchemeHandler, 200
 Session, 209
 SessionHandler, 216
 User, 229
 UserHandler, 237
stack_dump
 OWLException, 183
stat
 OWL, 180
StatusHandler, 219
 \$code, 221
 \$instance, 221
 \$params, 221
 __clone, 220
 __construct, 220
 get_code, 220
 get_instance, 220
 get_message, 220
 get_severity, 220
 reset, 221
 set_code, 221
 set_params, 221
Storage Object modules, 11
switch
 index.php, 242
table_description
 SchemeHandler, 200
table_exists
 DbHandler, 48
tablelist
 DbHandler, 48
tablename
 DbHandler, 49
traceback
 OWL, 17
 BaseElement, 24
 DataHandler, 36
 DbHandler, 49
 Dispatcher, 57
 FileHandler, 65
 Form, 74
 FormField, 82
 FormFieldButton, 92

FormFieldCheckbox, 101
 FormFieldFile, 109
 FormFieldRadio, 119
 FormFieldSelect, 128
 FormFieldText, 138
 FormFieldTextarea, 148
 FormHandler, 156
 ImageHandler, 164
 LogHandler, 172
 OWL, 180
 OWLEception, 183
 SchemeHandler, 200
 Session, 209
 SessionHandler, 217
 User, 229
 UserHandler, 237

 unescape_string
 DbHandler, 49
 update_list
 DbHandler, 49
 User, 223
 \$dataset, 230
 \$session, 230
 \$severity, 230
 \$user_data, 230
 __construct, 225
 __destruct, 225
 check, 226
 get_id, 226
 get_session_var, 226
 get_severity, 226
 get_status, 226
 get_username, 227
 init, 227
 login, 227
 logout, 227
 reset, 227
 set_high_severity, 228
 set_session_var, 228
 set_status, 228
 set_username, 229
 signal, 229
 traceback, 229
 UserHandler, 231
 \$dataset, 238
 \$session, 238
 \$severity, 238
 \$user_data, 238
 __construct, 233
 __destruct, 233
 check, 234
 get_severity, 234
 get_status, 234

 get_username, 234
 hash_password, 235
 init, 235
 login, 235
 logout, 235
 read_userdata, 236
 reset, 236
 set_high_severity, 236
 set_status, 236
 set_username, 237
 signal, 237
 traceback, 237

 validate_scheme
 SchemeHandler, 201

 where_clause
 DbHandler, 50
 write
 DbHandler, 50
 Session, 209
 SessionHandler, 217
 write_logfile
 LogHandler, 173