

general_ledger

Generated by Doxygen 1.8.1.2

Sun Jun 22 2014 23:46:00

Contents

1	General Ledger.	1
2	Todo List	3
3	Bug List	5
4	Module Index	7
4.1	Modules	7
5	Class Index	9
5.1	Class Hierarchy	9
6	Class Index	11
6.1	Class List	11
7	File Index	13
7.1	File List	13
8	Module Documentation	17
8.1	Program configuration module	17
8.1.1	Detailed Description	17
8.2	Database interaction module	18
8.2.1	Detailed Description	19
8.2.2	Function Documentation	19
8.2.2.1	get_connection	19
8.2.2.2	get_database_type	19
8.2.2.3	get_field_names	19
8.2.2.4	get_row	19
8.3	SQL statements module	21
8.3.1	Detailed Description	21
8.4	General Ledger database module.	22
8.4.1	Detailed Description	22
8.4.2	Function Documentation	23
8.4.2.1	decorated_report_from_table	23

8.4.2.2	decorated_row	23
8.4.2.3	grow_widths	23
8.4.2.4	max_column_widths	23
8.4.2.5	plain_report_from_table	24
8.4.2.6	plain_row	24
8.4.2.7	separator_row	24
8.5	Terminal program library	25
8.5.1	Detailed Description	25
8.6	General purpose utilities.	26
8.6.1	Detailed Description	27
8.6.2	Function Documentation	27
8.6.2.1	content_lines	27
8.6.2.2	currency_from_string	27
8.6.2.3	join	27
8.6.2.4	next_content_line	27
8.6.2.5	operator!=	28
8.6.2.6	operator+	28
8.6.2.7	operator-	28
8.6.2.8	operator<	28
8.6.2.9	operator<=	29
8.6.2.10	operator==	29
8.6.2.11	operator>	29
8.6.2.12	operator>=	30
8.6.2.13	replace	30
8.6.2.14	split	30
8.6.2.15	split	30
8.6.2.16	split_lines	31
8.6.2.17	trim	31
8.6.2.18	trim_back	31
8.6.2.19	trim_front	31
8.7	Database program.	33
8.7.1	Detailed Description	33
8.7.2	Function Documentation	33
8.7.2.1	check_db_parameters	33
8.7.2.2	check_help_and_version	33
8.7.2.3	login	34
8.7.2.4	main	34
8.7.2.5	set_configuration	34
8.8	Reporting program.	35
8.8.1	Detailed Description	35

8.8.2	Function Documentation	35
8.8.2.1	check_db_parameters	35
8.8.2.2	check_help_and_version	35
8.8.2.3	login	36
8.8.2.4	main	36
8.8.2.5	set_configuration	36
8.9	Terminal access program.	37
8.9.1	Detailed Description	37
8.9.2	Function Documentation	37
8.9.2.1	check_db_parameters	37
8.9.2.2	check_help_and_version	37
8.9.2.3	login	38
8.9.2.4	main	38
8.9.2.5	set_configuration	38
8.10	User administration program.	39
8.10.1	Detailed Description	39
8.10.2	Function Documentation	39
8.10.2.1	check_db_parameters	39
8.10.2.2	check_help_and_version	40
8.10.2.3	check_user_password	40
8.10.2.4	enable_user	40
8.10.2.5	get_user	40
8.10.2.6	login	40
8.10.2.7	main	41
8.10.2.8	set_configuration	41
8.10.2.9	set_user_password	41
8.10.2.10	show_user_details	41
9	Class Documentation	43
9.1	genleg::Config Class Reference	43
9.1.1	Detailed Description	43
9.1.2	Constructor & Destructor Documentation	43
9.1.2.1	Config	43
9.1.2.2	~Config	44
9.1.3	Member Function Documentation	44
9.1.3.1	add_cmdline_option	44
9.1.3.2	is_set	44
9.1.3.3	operator[]	44
9.1.3.4	populate_from_cmdline	44
9.1.3.5	populate_from_file	45

9.1.4	Member Data Documentation	45
9.1.4.1	m_opts_set	45
9.1.4.2	m_opts_supp	45
9.2	genleg::ConfigBadConfigFile Class Reference	45
9.2.1	Detailed Description	46
9.2.2	Constructor & Destructor Documentation	46
9.2.2.1	ConfigBadConfigFile	46
9.3	genleg::ConfigBadOption Class Reference	47
9.3.1	Detailed Description	47
9.3.2	Constructor & Destructor Documentation	47
9.3.2.1	ConfigBadOption	48
9.4	genleg::ConfigCouldNotOpenFile Class Reference	48
9.4.1	Detailed Description	49
9.4.2	Constructor & Destructor Documentation	49
9.4.2.1	ConfigCouldNotOpenFile	49
9.5	genleg::ConfigException Class Reference	49
9.5.1	Detailed Description	49
9.5.2	Constructor & Destructor Documentation	50
9.5.2.1	ConfigException	50
9.6	genleg::ConfigOptionNotSet Class Reference	50
9.6.1	Detailed Description	51
9.6.2	Constructor & Destructor Documentation	51
9.6.2.1	ConfigOptionNotSet	51
9.7	pgutils::Currency Class Reference	51
9.7.1	Detailed Description	52
9.7.2	Constructor & Destructor Documentation	52
9.7.2.1	Currency	52
9.7.3	Member Function Documentation	52
9.7.3.1	expand	52
9.7.3.2	operator+=	52
9.7.3.3	operator-	52
9.7.3.4	operator-=	53
9.7.4	Friends And Related Function Documentation	53
9.7.4.1	operator+	53
9.7.4.2	operator<	53
9.7.4.3	operator==	53
9.7.5	Member Data Documentation	54
9.7.5.1	m_frac	54
9.7.5.2	m_int	54
9.8	pgutils::CurrencyException Class Reference	54

9.8.1	Detailed Description	54
9.8.2	Constructor & Destructor Documentation	54
9.8.2.1	CurrencyException	54
9.9	gldb::DBConn Class Reference	54
9.9.1	Detailed Description	55
9.9.2	Constructor & Destructor Documentation	56
9.9.2.1	DBConn	56
9.9.2.2	DBConn	56
9.9.2.3	DBConn	56
9.9.3	Member Function Documentation	56
9.9.3.1	last_auto_increment	56
9.9.3.2	operator=	56
9.9.3.3	operator=	56
9.9.3.4	query	56
9.9.3.5	select	56
9.9.4	Member Data Documentation	57
9.9.4.1	m_imp	57
9.10	gldb::DBConnCouldNotConnect Class Reference	57
9.10.1	Detailed Description	58
9.10.2	Constructor & Destructor Documentation	58
9.10.2.1	DBConnCouldNotConnect	58
9.11	gldb::DBConnCouldNotQuery Class Reference	58
9.11.1	Detailed Description	59
9.11.2	Constructor & Destructor Documentation	59
9.11.2.1	DBConnCouldNotQuery	59
9.12	gldb::DBConnDummy Class Reference	59
9.12.1	Detailed Description	60
9.12.2	Constructor & Destructor Documentation	60
9.12.2.1	DBConnDummy	61
9.12.2.2	DBConnDummy	61
9.12.2.3	~DBConnDummy	61
9.12.3	Member Function Documentation	61
9.12.3.1	operator=	61
9.12.3.2	query	61
9.12.3.3	select	61
9.13	gldb::DBConnException Class Reference	62
9.13.1	Detailed Description	62
9.13.2	Constructor & Destructor Documentation	62
9.13.2.1	DBConnException	62
9.14	gldb::DBConnImp Class Reference	62

9.14.1	Detailed Description	63
9.14.2	Constructor & Destructor Documentation	63
9.14.2.1	DBConnImp	63
9.14.2.2	~DBConnImp	63
9.14.3	Member Function Documentation	64
9.14.3.1	last_auto_increment	64
9.14.3.2	query	64
9.14.3.3	select	64
9.15	gldb::DBConnMySQL Class Reference	64
9.15.1	Detailed Description	66
9.15.2	Constructor & Destructor Documentation	66
9.15.2.1	DBConnMySQL	66
9.15.2.2	DBConnMySQL	66
9.15.2.3	DBConnMySQL	66
9.15.2.4	~DBConnMySQL	66
9.15.3	Member Function Documentation	66
9.15.3.1	last_auto_increment	66
9.15.3.2	operator=	67
9.15.3.3	operator=	67
9.15.3.4	query	67
9.15.3.5	select	67
9.15.4	Member Data Documentation	67
9.15.4.1	m_conn	67
9.15.4.2	mtx	67
9.16	genleg::DBSQLDummy Class Reference	68
9.16.1	Detailed Description	68
9.17	genleg::DBSQLMySQL Class Reference	68
9.17.1	Detailed Description	69
9.18	genleg::DBSQLStatements Class Reference	69
9.18.1	Detailed Description	71
9.18.2	Constructor & Destructor Documentation	71
9.18.2.1	DBSQLStatements	71
9.18.2.2	~DBSQLStatements	71
9.18.3	Member Function Documentation	71
9.18.3.1	account_by_name	71
9.18.3.2	create_table	71
9.18.3.3	create_view	72
9.18.3.4	currenttb	72
9.18.3.5	currenttb_by_entity	72
9.18.3.6	drop_table	72

9.18.3.7	drop_view	72
9.18.3.8	entity_by_id	73
9.18.3.9	entity_by_name	73
9.18.3.10	get_perms	73
9.18.3.11	grant	73
9.18.3.12	je_by_id	74
9.18.3.13	jelines_by_id	74
9.18.3.14	listusers	74
9.18.3.15	post_je	74
9.18.3.16	post_je_line	75
9.18.3.17	revoke	75
9.18.3.18	standing_data	75
9.18.3.19	update_user	75
9.18.3.20	user_by_id	76
9.18.3.21	user_by_username	76
9.19	genleg::GLAccount Class Reference	76
9.19.1	Detailed Description	77
9.19.2	Constructor & Destructor Documentation	77
9.19.2.1	GLAccount	77
9.19.3	Member Data Documentation	77
9.19.3.1	m_description	77
9.19.3.2	m_enabled	77
9.20	genleg::GLDatabase Class Reference	77
9.20.1	Detailed Description	79
9.20.2	Constructor & Destructor Documentation	79
9.20.2.1	GLDatabase	79
9.20.2.2	~GLDatabase	80
9.20.3	Member Function Documentation	80
9.20.3.1	backend	80
9.20.3.2	create_entity	80
9.20.3.3	create_structure	80
9.20.3.4	create_user	80
9.20.3.5	current_trial_balance_report	81
9.20.3.6	destroy_structure	81
9.20.3.7	get_account_by_name	81
9.20.3.8	get_entity_by_id	81
9.20.3.9	get_entity_by_name	82
9.20.3.10	get_je_by_id	82
9.20.3.11	get_standing_data	82
9.20.3.12	get_user_by_id	82

9.20.3.13	get_user_by_username	83
9.20.3.14	grant	83
9.20.3.15	je_report	83
9.20.3.16	list_users_report	83
9.20.3.17	load_sample_data	83
9.20.3.18	post_journal	84
9.20.3.19	report	84
9.20.3.20	revoke	84
9.20.3.21	standing_data_report	84
9.20.3.22	update_user	84
9.20.4	Member Data Documentation	84
9.20.4.1	m_dbc	85
9.20.4.2	m_sql	85
9.20.4.3	m_tables	85
9.20.4.4	m_views	85
9.21	genleg::GLDBException Class Reference	85
9.21.1	Detailed Description	85
9.21.2	Constructor & Destructor Documentation	85
9.21.2.1	GLDBException	85
9.22	genleg::GLDBTransaction Class Reference	86
9.22.1	Detailed Description	86
9.22.2	Constructor & Destructor Documentation	86
9.22.2.1	GLDBTransaction	86
9.22.2.2	~GLDBTransaction	87
9.22.3	Member Data Documentation	87
9.22.3.1	m_commit	87
9.22.3.2	m_dbc	87
9.23	genleg::GLEntity Class Reference	87
9.23.1	Detailed Description	88
9.23.2	Constructor & Destructor Documentation	88
9.23.2.1	GLEntity	88
9.23.3	Member Data Documentation	88
9.23.3.1	m_aggregate	88
9.23.3.2	m_enabled	88
9.23.3.3	m_name	88
9.23.3.4	m_parent	88
9.23.3.5	m_shortcode	88
9.24	genleg::GLJELine Class Reference	89
9.24.1	Detailed Description	89
9.24.2	Constructor & Destructor Documentation	89

9.24.2.1	GLJELine	89
9.24.3	Member Function Documentation	90
9.24.3.1	account	90
9.24.3.2	amount	90
9.24.4	Member Data Documentation	90
9.24.4.1	m_acct	90
9.24.4.2	m_amount	90
9.25	genleg::GLJournal Class Reference	90
9.25.1	Detailed Description	91
9.25.2	Constructor & Destructor Documentation	91
9.25.2.1	GLJournal	91
9.25.3	Member Function Documentation	91
9.25.3.1	begin	91
9.25.3.2	begin	92
9.25.3.3	end	92
9.25.3.4	end	92
9.25.4	Member Data Documentation	92
9.25.4.1	m_entity	92
9.25.4.2	m_id	92
9.25.4.3	m_lines	92
9.25.4.4	m_memo	92
9.25.4.5	m_period	92
9.25.4.6	m_source	93
9.25.4.7	m_user	93
9.25.4.8	m_year	93
9.26	genleg::GLReport Class Reference	93
9.26.1	Detailed Description	93
9.26.2	Constructor & Destructor Documentation	93
9.26.2.1	GLReport	93
9.26.3	Member Data Documentation	93
9.26.3.1	m_headers	93
9.26.3.2	m_report_text	94
9.26.3.3	m_title	94
9.27	genleg::GLStandingData Class Reference	94
9.27.1	Detailed Description	94
9.27.2	Constructor & Destructor Documentation	94
9.27.2.1	GLStandingData	94
9.27.3	Member Data Documentation	94
9.27.3.1	m_num_periods	94
9.27.3.2	m_period	95

9.27.3.3	m_year	95
9.28	genleg::GLUser Class Reference	95
9.28.1	Detailed Description	96
9.28.2	Constructor & Destructor Documentation	96
9.28.2.1	GLUser	96
9.28.2.2	~GLUser	97
9.28.3	Member Function Documentation	97
9.28.3.1	check_password	97
9.28.3.2	enabled	97
9.28.3.3	firstname	97
9.28.3.4	id	97
9.28.3.5	lastname	97
9.28.3.6	pass_hash	98
9.28.3.7	pass_salt	98
9.28.3.8	permissions	98
9.28.3.9	set_enabled	98
9.28.3.10	set_firstname	98
9.28.3.11	set_lastname	98
9.28.3.12	set_password	98
9.28.3.13	set_username	99
9.28.3.14	username	99
9.28.4	Member Data Documentation	99
9.28.4.1	m_enabled	99
9.28.4.2	m_firstname	99
9.28.4.3	m_id	99
9.28.4.4	m_lastname	99
9.28.4.5	m_pass_hash	99
9.28.4.6	m_pass_salt	99
9.28.4.7	m_perms	99
9.28.4.8	m_username	100
9.29	gldb::MySQLResult Class Reference	100
9.29.1	Detailed Description	100
9.29.2	Constructor & Destructor Documentation	100
9.29.2.1	MySQLResult	100
9.29.2.2	~MySQLResult	101
9.29.2.3	MySQLResult	101
9.29.2.4	MySQLResult	101
9.29.3	Member Function Documentation	101
9.29.3.1	num_fields	101
9.29.3.2	operator=	101

9.29.3.3	operator=	101
9.29.3.4	result	101
9.29.4	Member Data Documentation	101
9.29.4.1	m_num_fields	101
9.29.4.2	m_result	101
9.30	pgcourses::Point Struct Reference	102
9.30.1	Detailed Description	102
9.30.2	Member Data Documentation	102
9.30.2.1	x	102
9.30.2.2	y	102
9.31	pgcourses::Rectangle Struct Reference	102
9.31.1	Detailed Description	103
9.31.2	Member Data Documentation	103
9.31.2.1	origin	103
9.31.2.2	size	103
9.32	pgcourses::Size Struct Reference	103
9.32.1	Detailed Description	104
9.32.2	Member Data Documentation	104
9.32.2.1	height	104
9.32.2.2	width	104
9.33	gldb::Table Class Reference	104
9.33.1	Detailed Description	106
9.33.2	Constructor & Destructor Documentation	106
9.33.2.1	Table	106
9.33.2.2	Table	106
9.33.2.3	Table	106
9.33.2.4	Table	106
9.33.2.5	~Table	106
9.33.3	Member Function Documentation	107
9.33.3.1	append_record	107
9.33.3.2	append_record	107
9.33.3.3	begin	107
9.33.3.4	begin	107
9.33.3.5	create_from_file	107
9.33.3.6	end	108
9.33.3.7	end	108
9.33.3.8	get_field	108
9.33.3.9	get_headers	108
9.33.3.10	insert_query	108
9.33.3.11	num_fields	109

9.33.3.12 num_records	109
9.33.3.13 operator=	109
9.33.3.14 operator=	109
9.33.3.15 operator[]	109
9.33.3.16 set_quoted	110
9.33.3.17 set_quoted	110
9.33.4 Member Data Documentation	110
9.33.4.1 m_headers	110
9.33.4.2 m_quoted	110
9.33.4.3 m_records	110
9.34 glldb::TableBadInputFile Class Reference	110
9.34.1 Detailed Description	111
9.34.2 Constructor & Destructor Documentation	111
9.34.2.1 TableBadInputFile	111
9.35 glldb::TableCouldNotOpenInputFile Class Reference	112
9.35.1 Detailed Description	112
9.35.2 Constructor & Destructor Documentation	112
9.35.2.1 TableCouldNotOpenInputFile	113
9.36 glldb::TableException Class Reference	113
9.36.1 Detailed Description	113
9.36.2 Constructor & Destructor Documentation	113
9.36.2.1 TableException	114
9.37 glldb::TableField Class Reference	114
9.37.1 Detailed Description	115
9.37.2 Constructor & Destructor Documentation	115
9.37.2.1 TableField	115
9.37.2.2 TableField	115
9.37.2.3 TableField	116
9.37.2.4 TableField	116
9.37.2.5 TableField	116
9.37.2.6 ~TableField	116
9.37.3 Member Function Documentation	116
9.37.3.1 length	116
9.37.3.2 operator std::string	116
9.37.3.3 operator+=	116
9.37.3.4 operator+=	117
9.37.3.5 operator=	117
9.37.3.6 operator=	117
9.37.3.7 operator=	117
9.37.3.8 operator=	118

9.37.3.9	operator=	118
9.37.3.10	operator[]	118
9.37.3.11	operator[]	118
9.37.4	Friends And Related Function Documentation	118
9.37.4.1	operator<<	118
9.37.5	Member Data Documentation	119
9.37.5.1	m_data	119
9.38	gldb::TableMismatchedRecordLength Class Reference	119
9.38.1	Detailed Description	120
9.38.2	Constructor & Destructor Documentation	120
9.38.2.1	TableMismatchedRecordLength	120
9.39	gldb::TableNoSuchField Class Reference	120
9.39.1	Detailed Description	121
9.39.2	Constructor & Destructor Documentation	121
9.39.2.1	TableNoSuchField	121
9.40	gldb::TableNoSuchRecord Class Reference	122
9.40.1	Detailed Description	122
9.40.2	Constructor & Destructor Documentation	122
9.40.2.1	TableNoSuchRecord	123
9.41	gldb::TableRow Class Reference	123
9.41.1	Detailed Description	124
9.41.2	Constructor & Destructor Documentation	124
9.41.2.1	TableRow	124
9.41.2.2	TableRow	124
9.41.2.3	TableRow	124
9.41.2.4	TableRow	124
9.41.2.5	TableRow	125
9.41.2.6	TableRow	125
9.41.2.7	TableRow	125
9.41.2.8	~TableRow	125
9.41.3	Member Function Documentation	125
9.41.3.1	append_field	125
9.41.3.2	append_field	125
9.41.3.3	append_field	126
9.41.3.4	append_field	126
9.41.3.5	append_field	126
9.41.3.6	begin	126
9.41.3.7	begin	126
9.41.3.8	end	126
9.41.3.9	end	126

9.41.3.10 operator=	127
9.41.3.11 operator=	127
9.41.3.12 operator[]	127
9.41.3.13 operator[]	127
9.41.3.14 print	128
9.41.3.15 record_string	128
9.41.3.16 record_string	128
9.41.3.17 size	128
9.41.4 Member Data Documentation	128
9.41.4.1 m_fields	128
9.42 pgcourses::TermProgram Class Reference	128
9.42.1 Detailed Description	129
9.42.2 Constructor & Destructor Documentation	129
9.42.2.1 TermProgram	129
9.42.2.2 ~TermProgram	129
9.42.2.3 TermProgram	129
9.42.2.4 TermProgram	129
9.42.3 Member Function Documentation	129
9.42.3.1 operator=	129
9.42.3.2 operator=	129
9.42.3.3 sleep	130
9.42.3.4 terminal_size	130
9.42.4 Member Data Documentation	130
9.42.4.1 m_imp	130
9.43 pgcourses::TermProgramImp Class Reference	130
9.43.1 Detailed Description	131
9.43.2 Constructor & Destructor Documentation	131
9.43.2.1 TermProgramImp	131
9.43.2.2 ~TermProgramImp	131
9.43.2.3 TermProgramImp	131
9.43.2.4 TermProgramImp	131
9.43.3 Member Function Documentation	131
9.43.3.1 operator=	131
9.43.3.2 operator=	131
9.43.3.3 sleep	132
9.43.3.4 terminal_size	132
9.43.4 Member Data Documentation	132
9.43.4.1 m_mainwin	132
9.43.4.2 m_old_cursor	132
9.43.4.3 m_termsize	132

9.44	pgcurses::TPEXception Class Reference	132
9.44.1	Detailed Description	132
9.44.2	Constructor & Destructor Documentation	133
9.44.2.1	TPEXception	133
9.45	pgcurses::TPWindow Class Reference	133
9.45.1	Detailed Description	133
9.45.2	Constructor & Destructor Documentation	134
9.45.2.1	TPWindow	134
9.45.2.2	TPWindow	134
9.45.2.3	~TPWindow	134
9.45.2.4	TPWindow	134
9.45.2.5	TPWindow	134
9.45.3	Member Function Documentation	134
9.45.3.1	operator=	134
9.45.3.2	operator=	134
9.45.3.3	write_char	134
9.45.3.4	write_char	135
9.45.3.5	write_str	135
9.45.3.6	write_str	135
9.45.4	Member Data Documentation	135
9.45.4.1	m_imp	135
9.46	pgcurses::TPWindowImp Class Reference	135
9.46.1	Detailed Description	136
9.46.2	Constructor & Destructor Documentation	137
9.46.2.1	TPWindowImp	137
9.46.2.2	TPWindowImp	137
9.46.2.3	~TPWindowImp	137
9.46.2.4	TPWindowImp	137
9.46.2.5	TPWindowImp	137
9.46.3	Member Function Documentation	137
9.46.3.1	operator=	137
9.46.3.2	operator=	137
9.46.3.3	write_char	137
9.46.3.4	write_char	138
9.46.3.5	write_str	138
9.46.3.6	write_str	138
9.46.4	Member Data Documentation	138
9.46.4.1	m_origin	138
9.46.4.2	m_size	138
9.46.4.3	m_win	138

10 File Documentation	139
10.1 lib/config/config.cpp File Reference	139
10.1.1 Detailed Description	139
10.2 lib/config/config.h File Reference	140
10.2.1 Detailed Description	141
10.3 lib/config/config_getopt.cpp File Reference	141
10.3.1 Detailed Description	141
10.3.2 Macro Definition Documentation	142
10.3.2.1 _XOPEN_SOURCE	142
10.4 lib/database/data_structures.h File Reference	142
10.4.1 Detailed Description	143
10.5 lib/database/database.h File Reference	143
10.5.1 Detailed Description	144
10.6 lib/database/dbconn.cpp File Reference	145
10.6.1 Detailed Description	145
10.7 lib/database/dbconn.h File Reference	146
10.7.1 Detailed Description	147
10.8 lib/database/dbconnimp.h File Reference	147
10.8.1 Detailed Description	149
10.9 lib/database/table.cpp File Reference	149
10.9.1 Detailed Description	149
10.10 lib/database/table.h File Reference	150
10.10.1 Detailed Description	151
10.11 lib/database/tablefield.cpp File Reference	151
10.11.1 Detailed Description	152
10.12 lib/database/tablefield.h File Reference	152
10.12.1 Detailed Description	153
10.13 lib/database/ablerow.cpp File Reference	153
10.13.1 Detailed Description	154
10.14 lib/database/ablerow.h File Reference	154
10.14.1 Detailed Description	155
10.15 lib/database_imp/database_imp.h File Reference	156
10.15.1 Detailed Description	157
10.16 lib/database_imp/dummy/dbconn_dummy_imp.cpp File Reference	157
10.16.1 Detailed Description	158
10.17 lib/database_imp/dummy/dbconn_dummy_imp.h File Reference	158
10.17.1 Detailed Description	160
10.18 lib/database_imp/mysql/dbconn_mysql_functions.cpp File Reference	160
10.18.1 Detailed Description	161
10.19 lib/database_imp/mysql/dbconn_mysql_imp.cpp File Reference	161

10.19.1 Detailed Description	162
10.20lib/database_imp/mysql/dbconn_mysql_imp.h File Reference	163
10.20.1 Detailed Description	164
10.21lib/database_imp/mysql/dbconn_mysql_result.cpp File Reference	164
10.21.1 Detailed Description	165
10.22lib/database_imp/mysql/dbconn_mysql_result.h File Reference	165
10.22.1 Detailed Description	166
10.23lib/dbsql/dbsql.h File Reference	166
10.23.1 Detailed Description	167
10.24lib/dbsql/dbsql_dummy.h File Reference	167
10.24.1 Detailed Description	168
10.25lib/dbsql/dbsql_functions.h File Reference	169
10.25.1 Detailed Description	170
10.26lib/dbsql/dbsql_implementations.h File Reference	170
10.26.1 Detailed Description	171
10.27lib/dbsql/dbsql_mysql.h File Reference	172
10.27.1 Detailed Description	173
10.28lib/dbsql/dbsqlstatements.cpp File Reference	173
10.28.1 Detailed Description	173
10.29lib/dbsql/dbsqlstatements.h File Reference	174
10.29.1 Detailed Description	175
10.30lib/gldb/glaccount.cpp File Reference	175
10.30.1 Detailed Description	175
10.31lib/gldb/glaccount.h File Reference	175
10.31.1 Detailed Description	176
10.32lib/gldb/gldatabase.cpp File Reference	177
10.32.1 Detailed Description	177
10.32.2 Function Documentation	177
10.32.2.1 boolstring_to_bool	178
10.33lib/gldb/gldatabase.h File Reference	178
10.33.1 Detailed Description	179
10.34lib/gldb/gldb.h File Reference	179
10.34.1 Detailed Description	180
10.35lib/gldb/glentity.cpp File Reference	180
10.35.1 Detailed Description	181
10.36lib/gldb/glentity.h File Reference	181
10.36.1 Detailed Description	182
10.37lib/gldb/glexception.h File Reference	183
10.37.1 Detailed Description	183
10.38lib/gldb/gljournal.cpp File Reference	184

10.38.1 Detailed Description	184
10.39lib/gldb/gljournal.h File Reference	184
10.39.1 Detailed Description	186
10.40lib/gldb/glreport.cpp File Reference	186
10.40.1 Detailed Description	187
10.41lib/gldb/glreport.h File Reference	187
10.41.1 Detailed Description	189
10.42lib/gldb/glstanding.cpp File Reference	189
10.42.1 Detailed Description	189
10.43lib/gldb/glstanding.h File Reference	190
10.43.1 Detailed Description	191
10.44lib/gldb/gluser.cpp File Reference	191
10.44.1 Detailed Description	191
10.45lib/gldb/gluser.h File Reference	191
10.45.1 Detailed Description	192
10.46lib/gldb/gluser_pass.cpp File Reference	193
10.46.1 Detailed Description	193
10.46.2 Macro Definition Documentation	194
10.46.2.1 _XOPEN_SOURCE	194
10.46.3 Function Documentation	194
10.46.3.1 generate_salt	194
10.47lib/pgcurses/pgcurses.h File Reference	194
10.47.1 Detailed Description	195
10.48lib/pgcurses/termprogram.cpp File Reference	195
10.48.1 Detailed Description	196
10.49lib/pgcurses/termprogram.h File Reference	196
10.49.1 Detailed Description	197
10.50lib/pgcurses/termprogramimp.cpp File Reference	197
10.50.1 Detailed Description	198
10.51lib/pgcurses/termprogramimp.h File Reference	198
10.51.1 Detailed Description	199
10.52lib/pgcurses/tpexception.h File Reference	199
10.52.1 Detailed Description	200
10.53lib/pgcurses/tpgentypes.h File Reference	200
10.53.1 Detailed Description	201
10.54lib/pgcurses/tpwindow.cpp File Reference	201
10.54.1 Detailed Description	201
10.55lib/pgcurses/tpwindow.h File Reference	202
10.55.1 Detailed Description	203
10.56lib/pgcurses/tpwindowimp.cpp File Reference	203

10.56.1 Detailed Description	203
10.57lib/pgcurses/tpwindowimp.h File Reference	204
10.57.1 Detailed Description	204
10.58lib/pgutils/currency.cpp File Reference	205
10.58.1 Detailed Description	205
10.59lib/pgutils/currency.h File Reference	205
10.59.1 Detailed Description	207
10.60lib/pgutils/pgutils.h File Reference	207
10.60.1 Detailed Description	208
10.61lib/pgutils/stringhelp.cpp File Reference	208
10.61.1 Detailed Description	209
10.62lib/pgutils/stringhelp.h File Reference	209
10.62.1 Detailed Description	210
10.63progs/gl_db/gl_db_main.cpp File Reference	210
10.63.1 Detailed Description	211
10.64progs/gl_report/gl_report_main.cpp File Reference	212
10.64.1 Detailed Description	213
10.65progs/gl_term/gl_term_main.cpp File Reference	213
10.65.1 Detailed Description	214
10.66progs/gl_user/gl_user_main.cpp File Reference	214
10.66.1 Detailed Description	216

Chapter 1

General Ledger.

General Ledger will be a fully-featured, multi-user, open-source general ledger system. The project is in the early stages of development.

Chapter 2

Todo List

File [gluser_pass.cpp](#)

Implement a better form of password encryption. In particular, these functions are not re-entrant, and only use the first 8 characters of the password.

Chapter 3

Bug List

Member `gldb::Table::Table` (`const Table &table`)

'explicit' removed from here after failure to compile at end of MySQL query function.

Chapter 4

Module Index

4.1 Modules

Here is a list of all modules:

Program configuration module	17
Database interaction module	18
SQL statements module	21
General Ledger database module.	22
Terminal program library	25
General purpose utilities.	26
Database program.	33
Reporting program.	35
Terminal access program.	37
User administration program.	39

Chapter 5

Class Index

5.1 Class Hierarchy

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

genleg::Config	43
genleg::ConfigException	49
genleg::ConfigBadConfigFile	45
genleg::ConfigBadOption	47
genleg::ConfigCouldNotOpenFile	48
genleg::ConfigOptionNotSet	50
pgutils::Currency	51
pgutils::CurrencyException	54
gldb::DBConn	54
gldb::DBConnException	62
gldb::DBConnCouldNotConnect	57
gldb::DBConnCouldNotQuery	58
gldb::DBConnImp	62
gldb::DBConnDummy	59
gldb::DBConnMySQL	64
genleg::DBSQLStatements	69
genleg::DBSQLDummy	68
genleg::DBSQLMySQL	68
genleg::GLAccount	76
genleg::GLDatabase	77
genleg::GLDBException	85
genleg::GLDBTransaction	86
genleg::GLEntity	87
genleg::GLJELine	89
genleg::GLJournal	90
genleg::GLReport	93
genleg::GLStandingData	94
genleg::GLUser	95
gldb::MySQLResult	100
pgcurses::Point	102
pgcurses::Rectangle	102
pgcurses::Size	103
gldb::Table	104
gldb::TableException	113
gldb::TableBadInputFile	110
gldb::TableCouldNotOpenInputFile	112
gldb::TableMismatchedRecordLength	119

gldb::TableNoSuchField	120
gldb::TableNoSuchRecord	122
gldb::TableField	114
gldb::TableRow	123
pgcourses::TermProgram	128
pgcourses::TermProgramImp	130
pgcourses::TPException	132
pgcourses::TPWindow	133
pgcourses::TPWindowImp	135

Chapter 6

Class Index

6.1 Class List

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

genleg::Config	Configuration options class	43
genleg::ConfigBadConfigFile	Exception class for badly formed configuration file	45
genleg::ConfigBadOption	Exception class for bad provided option	47
genleg::ConfigCouldNotOpenFile	Exception class for when conf file cannot be opened	48
genleg::ConfigException	Configuration module exception base class	49
genleg::ConfigOptionNotSet	Exception class for option not set	50
pgutils::Currency	Currency amount class	51
pgutils::CurrencyException	Base Currency exception class	54
gldb::DBConn	Database connection class	54
gldb::DBConnCouldNotConnect	Could not connect to database exception class	57
gldb::DBConnCouldNotQuery	Could not execute database query exception class	58
gldb::DBConnDummy	Dummy database implementation class	59
gldb::DBConnException	Base database connection exception class	62
gldb::DBConnImp	Abstract database implementation base class	62
gldb::DBConnMySQL	MySQL database implementation class	64
genleg::DBSQLDummy	Dummy SQL statements class	68
genleg::DBSQLMySQL	MySQL SQL statements class	68
genleg::DBSQLStatements	SQL statements class	69
genleg::GLAccount	Nominal account class	76

genleg::GLDatabase	
General ledger database class	77
genleg::GLDBException	
Base general ledger database exceptionc class	85
genleg::GLDBTransaction	
Database transaction RAll class	86
genleg::GLEntity	
General ledger entity class	87
genleg::GLJELine	
Journal entry line class	89
genleg::GLJournal	
Journal entry class	90
genleg::GLReport	
General ledger report class	93
genleg::GLStandingData	
General ledger standing data class	94
genleg::GLUser	
General ledger user class	95
glldb::MySQLResult	
MySQL result structure class	100
pgcurses::Point	
Point structure	102
pgcurses::Rectangle	
Rectangle structure	102
pgcurses::Size	
Size structure	103
glldb::Table	
Database table class	104
glldb::TableBadInputFile	
Could not connect to database exception class	110
glldb::TableCouldNotOpenInputFile	
Could not connect to database exception class	112
glldb::TableException	
Base database connection exception class	113
glldb::TableField	
Database table field class	114
glldb::TableMismatchedRecordLength	
Mismatched record length exception class	119
glldb::TableNoSuchField	
No such field exception class	120
glldb::TableNoSuchRecord	
No such record exception class	122
glldb::TableRow	
Database table row class	123
pgcurses::TermProgram	
Main terminal program class	128
pgcurses::TermProgramImp	
Main terminal program implementation class	130
pgcurses::TPException	
Base terminal program exception class	132
pgcurses::TPWindow	
Generic window class	133
pgcurses::TPWindowImp	
Terminal program window implementation class	135

Chapter 7

File Index

7.1 File List

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

lib/config/ config.cpp	
Implementation of program configurations class	139
lib/config/ config.h	
Interface to program configurations class	140
lib/config/ config_getopt.cpp	
Implementation of command line functionality	141
lib/database/ data_structures.h	
Main interface to database data structures	142
lib/database/ database.h	
User interface to database functionality	143
lib/database/ dbconn.cpp	
Implementation of database connection class	145
lib/database/ dbconn.h	
Interface to database connection base class	146
lib/database/ dbconnimp.h	
Interface to abstract database implementation base class	147
lib/database/ table.cpp	
Implementation of database table data structure	149
lib/database/ table.h	
Interface to database table data structure	150
lib/database/ tablefield.cpp	
Implementation of database table field class	151
lib/database/ tablefield.h	
Interface to database table field class	152
lib/database/ tablerow.cpp	
Implementation of database table row data structure	153
lib/database/ tablerow.h	
Interface to database table row data structure	154
lib/database_imp/ database_imp.h	
Interface to database implementation factory function	156
lib/database_imp/dummy/ dbconn_dummy_imp.cpp	
Implementation of Dummy database connection implementation class	157
lib/database_imp/dummy/ dbconn_dummy_imp.h	
Interface to dummy database connection implementation class	158
lib/database_imp/mysql/ dbconn_mysql_functions.cpp	
Implementation of MySQL implementation factory function	160
lib/database_imp/mysql/ dbconn_mysql_imp.cpp	
Implementation of MySQL database connection implementation class	161

lib/database_imp/mysql/dbconn_mysql_imp.h	
Interface to MySQL database connection implementation class	163
lib/database_imp/mysql/dbconn_mysql_result.cpp	
Implementation of MySQL result structure resource handle class	164
lib/database_imp/mysql/dbconn_mysql_result.h	
Interface to MySQL result structure resource handle class	165
lib/dbsql/dbsql.h	
User interface to DBSQL module	166
lib/dbsql/dbsql_dummy.h	
Interface to dummy SQL statement class	167
lib/dbsql/dbsql_functions.h	
Interface to SQL module standalone functions	169
lib/dbsql/dbsql_implementations.h	
Aggregation header for DBSqlStatements implementations	170
lib/dbsql/dbsql_mysql.h	
Interface to MySQL SQL statement class	172
lib/dbsql/dbsqlstatements.cpp	
Implementation of SQL statement class	173
lib/dbsql/dbsqlstatements.h	
Implementation of SQL module standalone functions	174
lib/gldb/glaccount.cpp	
Implementation of nominal account class	175
lib/gldb/glaccount.h	
Interface to nominal account class	175
lib/gldb/gldatabase.cpp	
Implementation of General Ledger database class	177
lib/gldb/gldatabase.h	
Interface to General Ledger database class	178
lib/gldb/gldb.h	
User interface to General Ledger database module	179
lib/gldb/glentity.cpp	
Implementation of general ledger entity class	180
lib/gldb/glentity.h	
Interface to general ledger entity class	181
lib/gldb/glexception.h	
Interface to General Ledger base exception class	183
lib/gldb/gljournal.cpp	
Implementation of journal entry classes	184
lib/gldb/gljournal.h	
Interface to journal entry classes	184
lib/gldb/glreport.cpp	
Implementation of report class	186
lib/gldb/glreport.h	
Interface to report class	187
lib/gldb/glstanding.cpp	
Implementation of general ledger standing data class	189
lib/gldb/glstanding.h	
Interface to general ledger standing data class	190
lib/gldb/gluser.cpp	
Implementation of user class	191
lib/gldb/gluser.h	
Interface to user class	191
lib/gldb/gluser_pass.cpp	
Implementation of password functions for user class	193
lib/pgcurses/pgcurses.h	
User interface to curses wrapper library	194
lib/pgcurses/termprogram.cpp	
Implementation of main terminal program class	195

lib/pgcurses/termprogram.h	
Interface to main terminal program class	196
lib/pgcurses/termprogramimp.cpp	
Implementation of main terminal program implementation class	197
lib/pgcurses/termprogramimp.h	
Interface to main terminal program implementation class	198
lib/pgcurses/tpexception.h	
Interface to terminal program exception class	199
lib/pgcurses/tpgentypes.h	
Interface to terminal program general types	200
lib/pgcurses/tpwindow.cpp	
Implementation of generic curses window class	201
lib/pgcurses/tpwindow.h	
Interface to generic curses window class	202
lib/pgcurses/tpwindowimp.cpp	
Implementation of generic curses window impl class	203
lib/pgcurses/tpwindowimp.h	
Interface to generic curses window impl class	204
lib/pgutils/currency.cpp	
Implementation of currency amount class	205
lib/pgutils/currency.h	
Interface to currency amount class	205
lib/pgutils/pgutils.h	
Aggregate interface to general utility functions	207
lib/pgutils/stringhelp.cpp	
Implementation of string helper functions	208
lib/pgutils/stringhelp.h	
Interface to string helper functions	209
progs/gl_db/gl_db_main.cpp	
Main functionality for gl_db program	210
progs/gl_report/gl_report_main.cpp	
Main functionality for gl_report program	212
progs/gl_term/gl_term_main.cpp	
Main functionality for gl_term program	213
progs/gl_user/gl_user_main.cpp	
Main functionality for gl_user program	214

Chapter 8

Module Documentation

8.1 Program configuration module

Classes

- class [genleg::ConfigException](#)
Configuration module exception base class.
- class [genleg::ConfigOptionNotSet](#)
Exception class for option not set.
- class [genleg::ConfigBadOption](#)
Exception class for bad provided option.
- class [genleg::ConfigCouldNotOpenFile](#)
Exception class for when conf file cannot be opened.
- class [genleg::ConfigBadConfigFile](#)
Exception class for badly formed configuration file.
- class [genleg::Config](#)
Configuration options class.

Enumerations

- enum [genleg::Argument](#)
Enumeration class for option argument specifications.

8.1.1 Detailed Description

Module for getting options from the command line and configuration files.

8.2 Database interaction module

Classes

- class [gldb::DBConnException](#)
Base database connection exception class.
- class [gldb::DBConnCouldNotConnect](#)
Could not connect to database exception class.
- class [gldb::DBConnCouldNotQuery](#)
Could not execute database query exception class.
- class [gldb::DBConn](#)
Database connection class.
- class [gldb::DBConnImp](#)
Abstract database implementation base class.
- class [gldb::TableException](#)
Base database connection exception class.
- class [gldb::TableNoSuchField](#)
No such field exception class.
- class [gldb::TableNoSuchRecord](#)
No such record exception class.
- class [gldb::TableMismatchedRecordLength](#)
Mismatched record length exception class.
- class [gldb::TableBadInputFile](#)
Could not connect to database exception class.
- class [gldb::TableCouldNotOpenInputFile](#)
Could not connect to database exception class.
- class [gldb::Table](#)
Database table class.
- class [gldb::TableField](#)
Database table field class.
- class [gldb::TableRow](#)
Database table row class.
- class [gldb::DBConnDummy](#)
Dummy database implementation class.
- class [gldb::DBConnMySQL](#)
MySQL database implementation class.
- class [gldb::MySQLResult](#)
MySQL result structure class.

Functions

- [DBConnImp * gldb::get_connection](#) (const std::string &database, const std::string &hostname, const std::string &username, const std::string &password)
Creates and returns a pointer to a database implementation.
- std::string [gldb::get_database_type](#) ()
Returns the name of the compiled-in database type.
- static [TableRow get_field_names](#) (MySQLResult &result)
Gets field names from a MySQL result structure.
- static [TableRow get_row](#) (MySQLResult &result, MYSQL_ROW row)
Creates a TableRow from a MySQL result row.

8.2.1 Detailed Description

Module for interacting with the database.

8.2.2 Function Documentation

8.2.2.1 DBConnImp * glldb::get_connection (const std::string & *database*, const std::string & *hostname*, const std::string & *username*, const std::string & *password*)

Creates and returns a pointer to a database implementation.

The implementation of this function is provided by the individual database implementations. One database implementation is compiled into the program at any one time. Multiple database systems are, or will be, supported, and not every system will possess the libraries and headers to compile every implementation. Therefore, only one implementation is compiled in at a time. The fact that each database implementation will implement this function to return the correct derived class prevents any attempt to compile unsupported library code. This would not be feasible if we were to simply provide each implementation as a subclass.

Parameters

<i>database</i>	The name of the database to which to connect.
<i>hostname</i>	The hostname of the computer running the database.
<i>username</i>	The username with which to log into the database.
<i>password</i>	The password with which to log into the database.

Returns

A pointer to the database implementation.

8.2.2.2 std::string glldb::get_database_type ()

Returns the name of the compiled-in database type.

Returns

The name of the compiled-in database type.

8.2.2.3 static TableRow get_field_names (MySQLResult & *result*) [static]

Gets field names from a MySQL result structure.

Parameters

<i>result</i>	The MySQL result structure.
---------------	-----------------------------

Returns

A TableRow containing the field names.

8.2.2.4 static TableRow get_row (MySQLResult & *result*, MYSQL_ROW *row*) [static]

Creates a TableRow from a MySQL result row.

Parameters

<i>result</i>	The MySQL result structure.
<i>row</i>	The MySQL row structure.

Returns

A TableRow containing the row data.

8.3 SQL statements module

Classes

- class [genleg::DBSQLDummy](#)
Dummy SQL statements class.
- class [genleg::DBSQLMySQL](#)
MySQL SQL statements class.
- class [genleg::DBSQLStatements](#)
SQL statements class.

8.3.1 Detailed Description

Module for producing SQL statements used by program.

8.4 General Ledger database module.

Classes

- class [genleg::GLAccount](#)
Nominal account class.
- class [genleg::GLDatabase](#)
General ledger database class.
- class [genleg::GLDBTransaction](#)
Database transaction RAIL class.
- class [genleg::GLEntity](#)
General ledger entity class.
- class [genleg::GLDBException](#)
Base general ledger database exception class.
- class [genleg::GLJELine](#)
Journal entry line class.
- class [genleg::GLJournal](#)
Journal entry class.
- class [genleg::GLReport](#)
General ledger report class.
- class [genleg::GLStandingData](#)
General ledger standing data class.
- class [genleg::GLUser](#)
General ledger user class.
- class [pgutils::CurrencyException](#)
Base [Currency](#) exception class.

Functions

- static `std::vector< size_t > max_column_widths` (const [gldb::Table](#) &table)
Calculates the maximum required column widths for a table.
- static void `grow_widths` (std::vector< size_t > &widths, const [TableRow](#) &row)
Increments a vector of required column widths.
- static `std::string separator_row` (const std::vector< size_t > &widths)
Returns a decorated separator row for a table.
- static `std::string plain_row` (const [TableRow](#) &row, const std::vector< size_t > &widths)
Returns a row for a plain report.
- static `std::string decorated_row` (const [TableRow](#) &row, const std::vector< size_t > &widths)
Returns a row for a decorated report.
- `std::string genleg::plain_report_from_table` (const [gldb::Table](#) &table)
Creates a plain report from a table.
- `std::string genleg::decorated_report_from_table` (const [gldb::Table](#) &table)
Creates a decorated report from a table.

8.4.1 Detailed Description

Module for interacting with the general ledger database model.

8.4.2 Function Documentation

8.4.2.1 `std::string genleg::decorated_report_from_table (const glldb::Table & table)`

Creates a decorated report from a table.

A "decorated report" presents the table surrounding with ASCII-art style lines consisting of '+' , '-' and '|' characters.

Parameters

<i>table</i>	The table from which to create the report.
--------------	--

Returns

A string containing the report.

8.4.2.2 `static std::string decorated_row (const TableRow & row, const std::vector< size_t > & widths) [static]`

Returns a row for a decorated report.

Parameters

<i>row</i>	The row for which to create the report row.
<i>widths</i>	A vector of required widths.

Returns

A string containing the decorated row.

8.4.2.3 `static void grow_widths (std::vector< size_t > & widths, const TableRow & row) [static]`

Increments a vector of required column widths.

Each element of the vector is increased to fit the width of each file in the row, if the existing width is not large enough to contain it.

Parameters

<i>widths</i>	An existing vector of widths.
<i>row</i>	The row against which to check and potentially increase the vector.

8.4.2.4 `static std::vector< size_t > max_column_widths (const glldb::Table & table) [static]`

Calculates the maximum required column widths for a table.

Parameters

<i>table</i>	The table.
--------------	------------

Returns

A vector of `size_t` containing the maximum required width for each column, without padding.

8.4.2.5 `std::string genleg::plain_report_from_table (const gldb::Table & table)`

Creates a plain report from a table.

A "plain report" separates each column with a space.

Parameters

<i>table</i>	The table from which to create the report.
--------------	--

Returns

A string containing the report.

8.4.2.6 `static std::string plain_row (const TableRow & row, const std::vector< size_t > & widths) [static]`

Returns a row for a plain report.

Parameters

<i>row</i>	The row for which to create the report row.
<i>widths</i>	A vector of required widths.

Returns

A string containing the plain row.

8.4.2.7 `static std::string separator_row (const std::vector< size_t > & widths) [static]`

Returns a decorated separator row for a table.

The "separator row" is of the format "+—+—+—+" where each column is separated by a '+' character, and consists of enough '-' characters to fit the respective width in the vector plus two additional characters for spacing.

Parameters

<i>widths</i>	A vector of required widths.
---------------	------------------------------

Returns

A string containing the separator row.

8.5 Terminal program library

Classes

- class `pgcurses::TermProgram`
Main terminal program class.
- class `pgcurses::TermProgramImp`
Main terminal program implementation class.
- class `pgcurses::TPException`
Base terminal program exception class.
- struct `pgcurses::Point`
Point structure.
- struct `pgcurses::Size`
Size structure.
- struct `pgcurses::Rectangle`
Rectangle structure.
- class `pgcurses::TPWindow`
Generic window class.
- class `pgcurses::TPWindowImp`
Terminal program window implementation class.

8.5.1 Detailed Description

Module for creating a terminal program.

8.6 General purpose utilities.

Classes

- class `pgutils::Currency`
Currency amount class.

Functions

- `Currency pgutils::operator+ (Currency lhs, const Currency &rhs)`
Currency addition operator.
- `Currency pgutils::operator- (Currency lhs, const Currency &rhs)`
Currency subtraction operator.
- `bool pgutils::operator== (const Currency &lhs, const Currency &rhs)`
Currency equality comparison operator.
- `bool pgutils::operator!= (const Currency &lhs, const Currency &rhs)`
Currency inequality comparison operator.
- `bool pgutils::operator< (const Currency &lhs, const Currency &rhs)`
Currency less than comparison operator.
- `bool pgutils::operator> (const Currency &lhs, const Currency &rhs)`
Currency greater than comparison operator.
- `bool pgutils::operator<= (const Currency &lhs, const Currency &rhs)`
Currency less than or equal to comparison operator.
- `bool pgutils::operator>= (const Currency &lhs, const Currency &rhs)`
Currency greater than or equal to comparison operator.
- `Currency pgutils::currency_from_string (const std::string &s)`
Creates a currency amount from a string representation.
- `std::string & pgutils::trim_front (std::string &s)`
Trims leading whitespace from a string.
- `std::string & pgutils::trim_back (std::string &s)`
Trims trailing whitespace from a string.
- `std::string & pgutils::trim (std::string &s)`
Trims leading and trailing whitespace from a string.
- `std::vector< std::string > pgutils::split (const std::string &s, const char delim)`
Splits a delimited string into tokens.
- `std::vector< std::string > & pgutils::split (std::vector< std::string > &vec, const std::string &s, const char delim)`
Splits a delimited string into tokens.
- `bool pgutils::next_content_line (std::istream &ifs, std::string &s)`
Gets the next content line from a stream.
- `std::vector< std::string > & pgutils::content_lines (std::vector< std::string > &vec, std::istream &ifs)`
Populates a vector of content lines from a stream.
- `std::vector< std::vector< std::string > > & pgutils::split_lines (std::vector< std::vector< std::string > > &vec, std::istream &ifs, const char delim)`
Populates a vector of vectors of fields from a stream.
- `std::string & pgutils::join (const std::vector< std::string > &vec, std::string &s, const char delim)`
Joins a vector of strings into a delimited line.
- `bool pgutils::replace (std::string &str, const std::string &from, const std::string &to)`
Replaces a substring with another string.

8.6.1 Detailed Description

General purpose utility classes and functions.

8.6.2 Function Documentation

8.6.2.1 `std::vector< std::string > & pgutils::content_lines (std::vector< std::string > & vec, std::istream & ifs)`

Populates a vector of content lines from a stream.

Parameters

<i>vec</i>	The vector to populate.
<i>ifs</i>	The input stream.

Returns

A reference to `vec`.

8.6.2.2 `pgutils::currency_from_string (const std::string & s)`

Creates a currency amount from a string representation.

Parameters

<i>s</i>	The string representation.
----------	----------------------------

Returns

The currency representation.

8.6.2.3 `std::string & pgutils::join (const std::vector< std::string > & vec, std::string & s, const char delim)`

Joins a vector of strings into a delimited line.

The function is the opposite of `split`.

Parameters

<i>vec</i>	The vector containing the strings.
<i>s</i>	The string in which to store the line.
<i>delim</i>	The delimiter character to be used to delimit the fields.

Returns

A reference to `s`.

8.6.2.4 `bool pgutils::next_content_line (std::istream & ifs, std::string & s)`

Gets the next content line from a stream.

A "content line" is defined as a line which, after being trimmed of trailing and leading whitespace, is not empty, and does not start with a '#' character (indicating a comment line).

Parameters

<i>ifs</i>	The input stream.
<i>s</i>	The string in which to store the line.

Returns

`true` if there is a next content line, `false` otherwise.

8.6.2.5 `bool pgutils::operator!=(const Currency & lhs, const Currency & rhs)`

[Currency](#) inequality comparison operator.

Parameters

<i>lhs</i>	Left hand side.
<i>rhs</i>	Right hand side.

Return values

<i>true</i>	If the two sides are not equal.
<i>false</i>	If the two sides are equal.

8.6.2.6 `Currency pgutils::operator+ (Currency lhs, const Currency & rhs)`

[Currency](#) addition operator.

Parameters

<i>lhs</i>	Left hand side.
<i>rhs</i>	Right hand side.

Returns

The sum of the two sides.

8.6.2.7 `Currency pgutils::operator- (Currency lhs, const Currency & rhs)`

[Currency](#) subtraction operator.

Parameters

<i>lhs</i>	Left hand side.
<i>rhs</i>	Right hand side.

Returns

The difference between the two sides.

8.6.2.8 `bool pgutils::operator< (const Currency & lhs, const Currency & rhs)`

[Currency](#) less than comparison operator.

Parameters

<i>lhs</i>	Left hand side.
<i>rhs</i>	Right hand side.

Return values

<i>true</i>	If the lhs is less than the rhs.
<i>false</i>	If the lhs is not less than the rhs.

8.6.2.9 `bool pgutils::operator<= (const Currency & lhs, const Currency & rhs)`

`Currency` less than or equal to comparison operator.

Parameters

<i>lhs</i>	Left hand side.
<i>rhs</i>	Right hand side.

Return values

<i>true</i>	If the lhs is less than or equal to the rhs.
<i>false</i>	If the lhs is not less than or equal to the rhs.

8.6.2.10 `bool pgutils::operator== (const Currency & lhs, const Currency & rhs)`

`Currency` equality comparison operator.

Parameters

<i>lhs</i>	Left hand side.
<i>rhs</i>	Right hand side.

Return values

<i>true</i>	If the two sides are equal.
<i>false</i>	If the two sides are not equal.

8.6.2.11 `bool pgutils::operator> (const Currency & lhs, const Currency & rhs)`

`Currency` greater than comparison operator.

Parameters

<i>lhs</i>	Left hand side.
<i>rhs</i>	Right hand side.

Return values

<i>true</i>	If the lhs is greater than the rhs.
<i>false</i>	If the lhs is not greater than the rhs.

8.6.2.12 `bool pgutils::operator>= (const Currency & lhs, const Currency & rhs)`

`Currency` greater than or equal to comparison operator.

Parameters

<i>lhs</i>	Left hand side.
<i>rhs</i>	Right hand side.

Return values

<i>true</i>	If the lhs is greater than or equal to the rhs.
<i>false</i>	If the lhs is not greater than or equal to the rhs.

8.6.2.13 `bool pgutils::replace (std::string & str, const std::string & from, const std::string & to)`

Replaces a substring with another string.

Parameters

<i>str</i>	The string containing the substring to replace.
<i>from</i>	The substring to replace.
<i>to</i>	The string with which to replace the substring.

Returns

`true` if a replacement was made, `false` otherwise.

8.6.2.14 `std::vector< std::string > pgutils::split (const std::string & s, const char delim)`

Splits a delimited string into tokens.

Parameters

<i>s</i>	The string to split.
<i>delim</i>	The delimiter character on which to split.

Returns

A vector of tokens.

8.6.2.15 `std::vector< std::string > & pgutils::split (std::vector< std::string > & vec, const std::string & s, const char delim)`

Splits a delimited string into tokens.

Parameters

<i>vec</i>	The vector into which to add the tokens.
<i>s</i>	The string to split.
<i>delim</i>	The delimiter character on which to split.

Returns

A reference to `vec`.

8.6.2.16 `std::vector< std::vector< std::string > > & pgutils::split_lines (std::vector< std::vector< std::string >> & vec, std::istream & ifs, const char delim)`

Populates a vector of vectors of fields from a stream.

Parameters

<i>vec</i>	The vector to populate.
<i>ifs</i>	The input stream.
<i>delim</i>	The delimiter character to split each content line.

Returns

A reference to `vec`.

8.6.2.17 `std::string & pgutils::trim (std::string & s)`

Trims leading and trailing whitespace from a string.

Parameters

<i>s</i>	The string to trim.
----------	---------------------

Returns

The trimmed string.

8.6.2.18 `std::string & pgutils::trim_back (std::string & s)`

Trims trailing whitespace from a string.

Parameters

<i>s</i>	The string to trim.
----------	---------------------

Returns

The trimmed string.

8.6.2.19 `std::string & pgutils::trim_front (std::string & s)`

Trims leading whitespace from a string.

Parameters

<i>s</i>	The string to trim.
----------	---------------------

Returns

The trimmed string.

8.7 Database program.

Functions

- static void `set_configuration` (`Config` &config, int argc, char *argv[])
Sets program configuration options.
- static bool `check_help_and_version` (const `Config` &config)
Prints help or version messages if requested.
- static bool `check_db_parameters` (const `Config` &config)
Checks if database, hostname and username were provided.
- static void `print_usage_message` ()
Prints a program usage message.
- static void `print_version_message` ()
Prints a program version message.
- static void `print_help_message` ()
Prints a program help message.
- static std::string `login` (void)
Gets a password from the terminal.
- int `main` (int argc, char *argv[])
Main function.

Variables

- static const char * `programe` = "gl_db"
Static variable for program name.

8.7.1 Detailed Description

Administrative database management program.

8.7.2 Function Documentation

8.7.2.1 static bool `check_db_parameters` (const `Config` & *config*) [static]

Checks if database, hostname and username were provided.

Parameters

<i>config</i>	Reference to a Config object.
---------------	-------------------------------

Returns

`true` if the information was provided, `false` otherwise.

8.7.2.2 static bool `check_help_and_version` (const `Config` & *config*) [static]

Prints help or version messages if requested.

Parameters

<i>config</i>	Reference to a Config object.
---------------	-------------------------------

Returns

`true` if the help or version message was requested, `false` otherwise.

8.7.2.3 static std::string login (void) [static]

Gets a password from the terminal.

Returns

The password.

8.7.2.4 int main (int argc, char * argv[])

Main function.

Parameters

<i>argc</i>	Number of command line arguments.
<i>argv</i>	Command line arguments.

Returns

Exit status code.

8.7.2.5 static void set_configuration (Config & config, int argc, char * argv[]) [static]

Sets program configuration options.

Parameters

<i>config</i>	Reference to a Config object.
<i>argc</i>	<code>argc</code> passed to <code>main()</code> .
<i>argv</i>	<code>argv</code> passed to <code>main()</code> .

8.8 Reporting program.

Functions

- static void `set_configuration` (`Config` &config, int argc, char *argv[])
Sets program configuration options.
- static bool `check_help_and_version` (const `Config` &config)
Prints help or version messages if requested.
- static bool `check_db_parameters` (const `Config` &config)
Checks if database, hostname and username were provided.
- static void `print_usage_message` ()
Prints a program usage message.
- static void `print_version_message` ()
Prints a program version message.
- static void `print_help_message` ()
Prints a program help message.
- static std::string `login` (void)
Gets a password from the terminal.
- int `main` (int argc, char *argv[])
Main function.

Variables

- static const char * `programe` = "gl_report"
Static variable for program name.

8.8.1 Detailed Description

Administrative reporting program.

8.8.2 Function Documentation

8.8.2.1 static bool `check_db_parameters` (const `Config` & *config*) [static]

Checks if database, hostname and username were provided.

Parameters

<i>config</i>	Reference to a Config object.
---------------	-------------------------------

Returns

`true` if the information was provided, `false` otherwise.

8.8.2.2 static bool `check_help_and_version` (const `Config` & *config*) [static]

Prints help or version messages if requested.

Parameters

<i>config</i>	Reference to a Config object.
---------------	-------------------------------

Returns

`true` if the help or version message was requested, `false` otherwise.

8.8.2.3 static std::string login (void) [static]

Gets a password from the terminal.

Returns

The password.

8.8.2.4 int main (int argc, char * argv[])

Main function.

Parameters

<i>argc</i>	Number of command line arguments.
<i>argv</i>	Command line arguments.

Returns

Exit status code.

8.8.2.5 static void set_configuration (Config & config, int argc, char * argv[]) [static]

Sets program configuration options.

Parameters

<i>config</i>	Reference to a Config object.
<i>argc</i>	<code>argc</code> passed to <code>main()</code> .
<i>argv</i>	<code>argv</code> passed to <code>main()</code> .

8.9 Terminal access program.

Functions

- static void `set_configuration` (`Config` &config, int argc, char *argv[])
Sets program configuration options.
- static bool `check_help_and_version` (const `Config` &config)
Prints help or version messages if requested.
- static bool `check_db_parameters` (const `Config` &config)
Checks if database, hostname and username were provided.
- static void `print_usage_message` ()
Prints a program usage message.
- static void `print_version_message` ()
Prints a program version message.
- static void `print_help_message` ()
Prints a program help message.
- static std::string `login` (void)
Gets a password from the terminal.
- int `main` (int argc, char *argv[])
Main function.

Variables

- static const char * `progrname` = "gl_term"
Static variable for program name.

8.9.1 Detailed Description

Terminal access program.

8.9.2 Function Documentation

8.9.2.1 static bool `check_db_parameters` (const `Config` & `config`) [static]

Checks if database, hostname and username were provided.

Parameters

<code>config</code>	Reference to a Config object.
---------------------	-------------------------------

Returns

`true` if the information was provided, `false` otherwise.

8.9.2.2 static bool `check_help_and_version` (const `Config` & `config`) [static]

Prints help or version messages if requested.

Parameters

<code>config</code>	Reference to a Config object.
---------------------	-------------------------------

Returns

`true` if the help or version message was requested, `false` otherwise.

8.9.2.3 static std::string login (void) [static]

Gets a password from the terminal.

Returns

The password.

8.9.2.4 int main (int argc, char * argv[])

Main function.

Parameters

<i>argc</i>	Number of command line arguments.
<i>argv</i>	Command line arguments.

Returns

Exit status code.

8.9.2.5 static void set_configuration (Config & config, int argc, char * argv[]) [static]

Sets program configuration options.

Parameters

<i>config</i>	Reference to a Config object.
<i>argc</i>	<code>argc</code> passed to <code>main()</code> .
<i>argv</i>	<code>argv</code> passed to <code>main()</code> .

8.10 User administration program.

Functions

- static void `set_configuration` (`Config` &config, int argc, char *argv[])
Sets program configuration options.
- static bool `check_help_and_version` (const `Config` &config)
Prints help or version messages if requested.
- static bool `check_db_parameters` (const `Config` &config)
Checks if database, hostname and username were provided.
- `GLUser` `get_user` (`Config` &config, `GLDatabase` &gdb)
Returns a user from either an ID or a name.
- static void `show_user_details` (const `GLUser` &user)
Outputs details for a user.
- static void `enable_user` (`GLUser` &user, `Config` &config, `GLDatabase` &gdb)
Enables or disables a user.
- static void `set_user_password` (`GLUser` &user, `Config` &config, `GLDatabase` &gdb)
Sets a user's password.
- static void `check_user_password` (`GLUser` &user, `Config` &config)
Checks a user's password.
- static void `print_usage_message` ()
Prints a program usage message.
- static void `print_version_message` ()
Prints a program version message.
- static void `print_help_message` ()
Prints a program help message.
- static std::string `login` (void)
Gets a password from the terminal.
- int `main` (int argc, char *argv[])
Main function.

Variables

- static const char * `progrname` = "gl_user"
Static variable for program name.

8.10.1 Detailed Description

User administration program.

8.10.2 Function Documentation

8.10.2.1 static bool check_db_parameters (const `Config` & config) [static]

Checks if database, hostname and username were provided.

Parameters

<code>config</code>	Reference to a <code>Config</code> object.
---------------------	--

Returns

`true` if the information was provided, `false` otherwise.

8.10.2.2 static bool check_help_and_version (const Config & config) [static]

Prints help or version messages if requested.

Parameters

<i>config</i>	Reference to a Config object.
---------------	-------------------------------

Returns

`true` if the help or version message was requested, `false` otherwise.

8.10.2.3 static void check_user_password (GLUser & user, Config & config) [static]

Checks a user's password.

Parameters

<i>user</i>	Reference to user.
<i>config</i>	Reference to program configuration options.

8.10.2.4 static void enable_user (GLUser & user, Config & config, GLDatabase & gdb) [static]

Enables or disables a user.

Parameters

<i>user</i>	Reference to user.
<i>config</i>	Reference to program configuration.
<i>gdb</i>	Reference to database object.

8.10.2.5 GLUser get_user (Config & config, GLDatabase & gdb)

Returns a user from either an ID or a name.

Parameters

<i>config</i>	Program configurations object.
<i>gdb</i>	Database object.

Returns

The user.

8.10.2.6 static std::string login (void) [static]

Gets a password from the terminal.

Returns

The password.

8.10.2.7 `int main (int argc, char * argv[])`

Main function.

Parameters

<i>argc</i>	Number of command line arguments.
<i>argv</i>	Command line arguments.

Returns

Exit status code.

8.10.2.8 `static void set_configuration (Config & config, int argc, char * argv[])` [static]

Sets program configuration options.

Parameters

<i>config</i>	Reference to a Config object.
<i>argc</i>	argc passed to <code>main()</code> .
<i>argv</i>	argv passed to <code>main()</code> .

8.10.2.9 `static void set_user_password (GLUser & user, Config & config, GLDatabase & gdb)` [static]

Sets a user's password.

Parameters

<i>user</i>	Reference to user.
<i>config</i>	Reference to program configuration.
<i>gdb</i>	Reference to database object.

8.10.2.10 `static void show_user_details (const GLUser & user)` [static]

Outputs details for a user.

Parameters

<i>user</i>	Reference to user.
-------------	--------------------

Chapter 9

Class Documentation

9.1 genleg::Config Class Reference

Configuration options class.

```
#include <config.h>
```

Public Member Functions

- [Config](#) ()
- [~Config](#) ()
- void [add_cmdline_option](#) (const std::string option, const enum [Argument](#) arg)
Adds a supported command line option.
- void [populate_from_cmdline](#) (const int argc, char *const *argv)
Populates options from the command line.
- void [populate_from_file](#) (const std::string filename)
Populates options from a configuration file.
- bool [is_set](#) (const std::string option) const
Checks if an option is set.
- const std::string & [operator\[\]](#) (const std::string &option) const
operator[] overload.

Private Attributes

- std::map< std::string,
std::string > [m_opts_set](#)
- std::list< std::pair
< std::string, enum [Argument](#) > > [m_opts_supp](#)

9.1.1 Detailed Description

Configuration options class.

9.1.2 Constructor & Destructor Documentation

9.1.2.1 Config::Config ()

Constructor

9.1.2.2 Config::~~Config ()

Destructor

9.1.3 Member Function Documentation

9.1.3.1 void Config::add_cmdline_option (const std::string *option*, const enum Argument *arg*)

Adds a supported command line option.

Parameters

<i>option</i>	The name of the option.
<i>arg</i>	The argument specification for the option.

9.1.3.2 bool Config::is_set (const std::string *option*) const

Checks is an option is set.

Parameters

<i>option</i>	The name of the option to check.
---------------	----------------------------------

Returns

`true` if the option has been set, `false` if it has not.

9.1.3.3 const std::string & Config::operator[] (const std::string & *option*) const

operator[] overload.

Retrieves the value of a set option.

Parameters

<i>option</i>	The name of the option.
---------------	-------------------------

Returns

The value of the option.

Exceptions

<i>ConfigOptionNotSet</i>	If the named option has not been set.
---	---------------------------------------

9.1.3.4 void Config::populate_from_cmdline (const int *argc*, char *const * *argv*)

Populates options from the command line.

Parameters

<i>argc</i>	<i>argc</i> supplied to <code>main()</code> .
<i>argv</i>	<i>argv</i> supplied to <code>main()</code> .

Exceptions

<i>ConfigBadOption</i>	If an unsupported option is specified, or if a required argument is missing, or if an unexpected argument is found.
--	---

9.1.3.5 void Config::populate_from_file (const std::string filename)

Populates options from a configuration file.

Parameters

<i>filename</i>	The name of the configuration file.
-----------------	-------------------------------------

Exceptions

<i>ConfigCouldNotOpenFile</i>	If the configuration file cannot be opened.
<i>ConfigBadConfigFile</i>	If the configuration file is badly formed.

9.1.4 Member Data Documentation

9.1.4.1 std::map<std::string, std::string> genleg::Config::m_opts_set [private]

Map of options which have been set

9.1.4.2 std::list<std::pair<std::string, enum Argument> > genleg::Config::m_opts_supp [private]

List of options which are supported

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

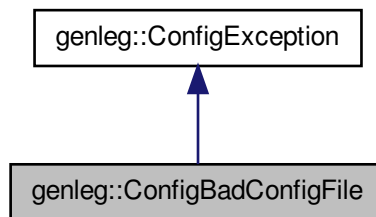
- lib/config/[config.h](#)
- lib/config/[config.cpp](#)
- lib/config/[config_getopt.cpp](#)

9.2 genleg::ConfigBadConfigFile Class Reference

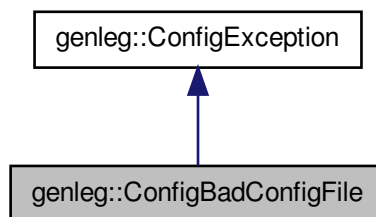
Exception class for badly formed configuration file.

```
#include <config.h>
```

Inheritance diagram for `genleg::ConfigBadConfigFile`:



Collaboration diagram for `genleg::ConfigBadConfigFile`:



Public Member Functions

- [ConfigBadConfigFile](#) (const std::string &msg)
Constructor.

9.2.1 Detailed Description

Exception class for badly formed configuration file.

9.2.2 Constructor & Destructor Documentation

9.2.2.1 `genleg::ConfigBadConfigFile::ConfigBadConfigFile (const std::string & msg) [inline], [explicit]`

Constructor.

Parameters

<i>msg</i>	Database error message
------------	------------------------

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

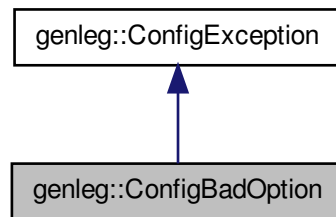
- [lib/config/config.h](#)

9.3 genleg::ConfigBadOption Class Reference

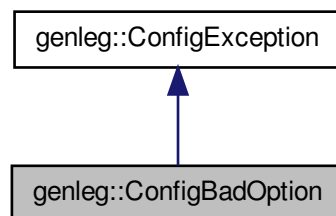
Exception class for bad provided option.

```
#include <config.h>
```

Inheritance diagram for genleg::ConfigBadOption:



Collaboration diagram for genleg::ConfigBadOption:



Public Member Functions

- [ConfigBadOption](#) (const std::string &msg)
Constructor.

9.3.1 Detailed Description

Exception class for bad provided option.

9.3.2 Constructor & Destructor Documentation

9.3.2.1 `genleg::ConfigBadOption::ConfigBadOption (const std::string & msg) [inline], [explicit]`

Constructor.

Parameters

<i>msg</i>	Database error message
------------	------------------------

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

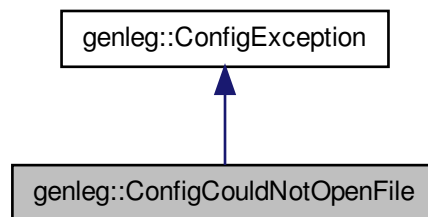
- [lib/config/config.h](#)

9.4 `genleg::ConfigCouldNotOpenFile` Class Reference

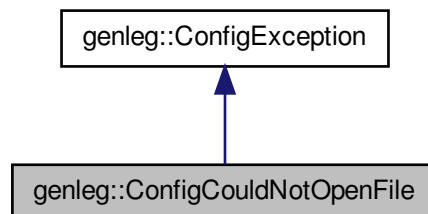
Exception class for when conf file cannot be opened.

```
#include <config.h>
```

Inheritance diagram for `genleg::ConfigCouldNotOpenFile`:



Collaboration diagram for `genleg::ConfigCouldNotOpenFile`:



Public Member Functions

- [ConfigCouldNotOpenFile](#) (const std::string &msg)
Constructor.

9.4.1 Detailed Description

Exception class for when conf file cannot be opened.

9.4.2 Constructor & Destructor Documentation

9.4.2.1 `genleg::ConfigCouldNotOpenFile::ConfigCouldNotOpenFile (const std::string & msg) [inline], [explicit]`

Constructor.

Parameters

<i>msg</i>	Database error message
------------	------------------------

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

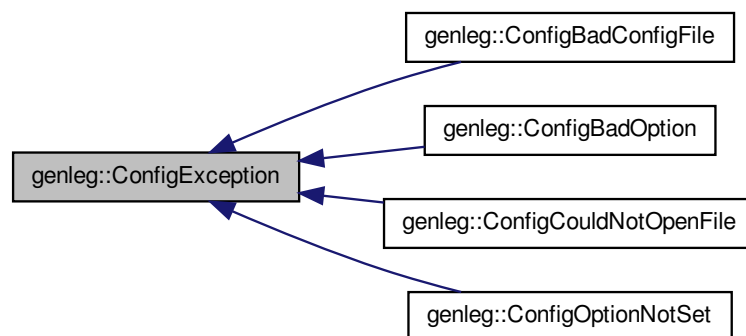
- [lib/config/config.h](#)

9.5 genleg::ConfigException Class Reference

Configuration module exception base class.

```
#include <config.h>
```

Inheritance diagram for genleg::ConfigException:



Public Member Functions

- [ConfigException](#) (const std::string &msg)
Constructor.

9.5.1 Detailed Description

Configuration module exception base class.

9.5.2 Constructor & Destructor Documentation

9.5.2.1 `genleg::ConfigException::ConfigException (const std::string & msg) [inline],[explicit]`

Constructor.

Parameters

<i>msg</i>	Database error message
------------	------------------------

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

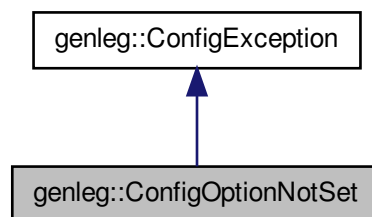
- `lib/config/config.h`

9.6 `genleg::ConfigOptionNotSet` Class Reference

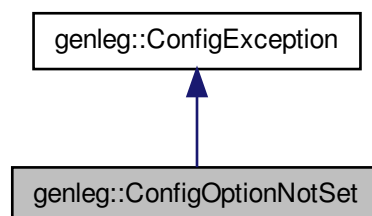
Exception class for option not set.

```
#include <config.h>
```

Inheritance diagram for `genleg::ConfigOptionNotSet`:



Collaboration diagram for `genleg::ConfigOptionNotSet`:



Public Member Functions

- [`ConfigOptionNotSet`](#) (const std::string &msg)

Constructor.

9.6.1 Detailed Description

Exception class for option not set.

9.6.2 Constructor & Destructor Documentation

9.6.2.1 `genleg::ConfigOptionNotSet::ConfigOptionNotSet (const std::string & msg) [inline],[explicit]`

Constructor.

Parameters

<code>msg</code>	Database error message
------------------	------------------------

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

- lib/config/[config.h](#)

9.7 pgutils::Currency Class Reference

[Currency](#) amount class.

```
#include <currency.h>
```

Public Member Functions

- [Currency](#) (const int64_t i=0, const uint8_t f=0)
Constructor.
- [Currency operator-](#) () const
Unary negation opertor.
- [Currency & operator+=](#) (const [Currency](#) &rhs)
Addition assignment operator.
- [Currency & operator-=](#) (const [Currency](#) &rhs)
Subtraction assignment operator.

Private Member Functions

- int64_t [expand](#) () const
Returns a [Currency](#) amount as a whole integer.

Private Attributes

- int64_t [m_int](#)
- int [m_frac](#)

Friends

- bool `operator==` (const [Currency](#) &lhs, const [Currency](#) &rhs)
[Currency](#) equality comparison operator.
- bool `operator<` (const [Currency](#) &lhs, const [Currency](#) &rhs)
[Currency](#) less than comparison operator.
- [Currency](#) `operator+` ([Currency](#) lhs, const [Currency](#) &rhs)
[Currency](#) addition operator.

9.7.1 Detailed Description

[Currency](#) amount class.

9.7.2 Constructor & Destructor Documentation

9.7.2.1 `pgutils::Currency::Currency (const int64_t i = 0, const uint8_t f = 0)` `[inline]`, `[explicit]`

Constructor.

Parameters

<i>i</i>	The integer part.
<i>f</i>	The fractional part.

9.7.3 Member Function Documentation

9.7.3.1 `int64_t pgutils::Currency::expand () const` `[inline]`, `[private]`

Returns a [Currency](#) amount as a whole integer.

Returns

The [Currency](#) amount expanded to a whole integer, equivalent to multiplying it by 100.

9.7.3.2 `Currency & Currency::operator+= (const Currency & rhs)`

Addition assignment operator.

Parameters

<i>rhs</i>	Right hand side currency amount.
------------	----------------------------------

Returns

A reference to the original currency amount.

9.7.3.3 `Currency pgutils::Currency::operator- () const` `[inline]`

Unary negation operator.

Returns

The negated currency amount.

9.7.3.4 Currency & Currency::operator-= (const Currency & rhs)

Subtraction assignment operator.

Parameters

<i>rhs</i>	Right hand side currency amount.
------------	----------------------------------

Returns

A reference to the original currency amount.

9.7.4 Friends And Related Function Documentation

9.7.4.1 Currency operator+ (Currency lhs, const Currency & rhs) [friend]

[Currency](#) addition operator.

Friend addition operator function

Parameters

<i>lhs</i>	Left hand side.
<i>rhs</i>	Right hand side.

Returns

The sum of the two sides.

9.7.4.2 bool operator< (const Currency & lhs, const Currency & rhs) [friend]

[Currency](#) less than comparison operator.

Friend less than comparison operator function

Parameters

<i>lhs</i>	Left hand side.
<i>rhs</i>	Right hand side.

Return values

<i>true</i>	If the lhs is less than the rhs.
<i>false</i>	If the lhs is not less than the rhs.

9.7.4.3 bool operator== (const Currency & lhs, const Currency & rhs) [friend]

[Currency](#) equality comparison operator.

Friend equality operator function

Parameters

<i>lhs</i>	Left hand side.
<i>rhs</i>	Right hand side.

Return values

<i>true</i>	If the two sides are equal.
<i>false</i>	If the two sides are not equal.

9.7.5 Member Data Documentation

9.7.5.1 `int pgutils::Currency::m_frac` `[private]`

Fractional part

9.7.5.2 `int64_t pgutils::Currency::m_int` `[private]`

Integer part

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

- [lib/pgutils/currency.h](#)
- [lib/pgutils/currency.cpp](#)

9.8 `pgutils::CurrencyException` Class Reference

Base [Currency](#) exception class.

```
#include <currency.h>
```

Public Member Functions

- [CurrencyException](#) (const std::string &msg)
Constructor.

9.8.1 Detailed Description

Base [Currency](#) exception class.

9.8.2 Constructor & Destructor Documentation

9.8.2.1 `pgutils::CurrencyException::CurrencyException (const std::string & msg)` `[inline]`, `[explicit]`

Constructor.

Parameters

<i>msg</i>	Database error message
------------	------------------------

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

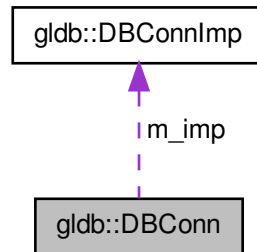
- [lib/pgutils/currency.h](#)

9.9 `glldb::DBConn` Class Reference

Database connection class.

```
#include <dbconn.h>
```

Collaboration diagram for glldb::DBConn:



Public Member Functions

- [DBConn](#) ([DBConnImp](#) *imp)
Constructor.
- [~DBConn](#) ()
Destructor..
- void [query](#) (const std::string &sql_query)
Runs an SQL query.
- [Table select](#) (const std::string &query)
Runs an SQL SELECT query.
- void [begin_transaction](#) ()
Begins a transaction.
- void [rollback_transaction](#) ()
Rolls back a transaction.
- void [commit_transaction](#) ()
Commits a transaction.
- unsigned long long [last_auto_increment](#) ()
Returns the last auto incremented value.
- [DBConn](#) (const [DBConn](#) &)
- [DBConn](#) (const [DBConn](#) &&)
- [DBConn](#) & [operator=](#) (const [DBConn](#) &)
- [DBConn](#) & [operator=](#) (const [DBConn](#) &&)

Private Attributes

- [DBConnImp](#) * [m_imp](#)

9.9.1 Detailed Description

Database connection class.

9.9.2 Constructor & Destructor Documentation

9.9.2.1 DBConn::DBConn (DBConnImp * *imp*) [explicit]

Constructor.

Parameters

<i>imp</i>	Pointer to database implementation object.
------------	--

9.9.2.2 glldb::DBConn::DBConn (const DBConn &)

Deleted copy constructor

9.9.2.3 glldb::DBConn::DBConn (const DBConn &&)

Deleted move constructor

9.9.3 Member Function Documentation

9.9.3.1 unsigned long long DBConn::last_auto_increment ()

Returns the last auto incremented value.

Returns

The last auto incremented value.

9.9.3.2 DBConn& glldb::DBConn::operator= (const DBConn &)

Deleted copy assignment operator

9.9.3.3 DBConn& glldb::DBConn::operator= (const DBConn &&)

Deleted move assignment operator

9.9.3.4 void DBConn::query (const std::string & *sql_query*)

Runs an SQL query.

Parameters

<i>sql_query</i>	The query.
------------------	------------

Returns

A [Table](#) object containing the results.

9.9.3.5 Table DBConn::select (const std::string & *query*)

Runs an SQL SELECT query.

Parameters

<i>query</i>	The query.
--------------	------------

Returns

A [Table](#) object containing the results.

9.9.4 Member Data Documentation

9.9.4.1 DBConnImp* glldb::DBConn::m_imp [private]

Pointer to database implementation object.

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

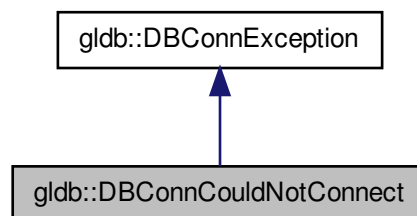
- lib/database/dbconn.h
- lib/database/dbconn.cpp

9.10 glldb::DBConnCouldNotConnect Class Reference

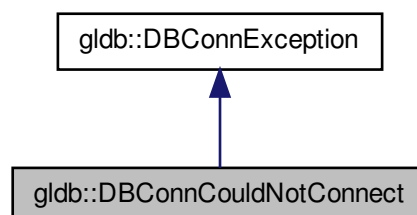
Could not connect to database exception class.

```
#include <dbconn.h>
```

Inheritance diagram for glldb::DBConnCouldNotConnect:



Collaboration diagram for glldb::DBConnCouldNotConnect:



Public Member Functions

- [DBConnCouldNotConnect](#) (const std::string &msg)
Constructor.

9.10.1 Detailed Description

Could not connect to database exception class.

9.10.2 Constructor & Destructor Documentation

9.10.2.1 `gldb::DBConnCouldNotConnect::DBConnCouldNotConnect (const std::string & msg) [inline], [explicit]`

Constructor.

Parameters

<i>msg</i>	Database error message
------------	------------------------

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

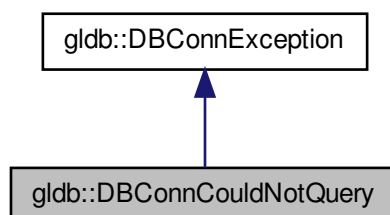
- lib/database/[dbconn.h](#)

9.11 gldb::DBConnCouldNotQuery Class Reference

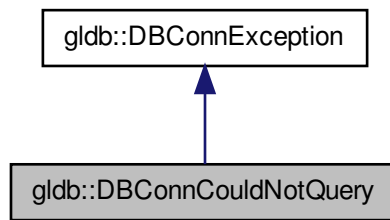
Could not execute database query exception class.

```
#include <dbconn.h>
```

Inheritance diagram for gldb::DBConnCouldNotQuery:



Collaboration diagram for glldb::DBConnCouldNotQuery:



Public Member Functions

- [DBConnCouldNotQuery](#) (const std::string &msg)
Constructor.

9.11.1 Detailed Description

Could not execute database query exception class.

9.11.2 Constructor & Destructor Documentation

9.11.2.1 `glldb::DBConnCouldNotQuery::DBConnCouldNotQuery (const std::string & msg)` `[inline], [explicit]`

Constructor.

Parameters

<i>msg</i>	Database error message
------------	------------------------

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

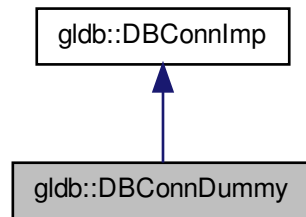
- lib/database/[dbconn.h](#)

9.12 glldb::DBConnDummy Class Reference

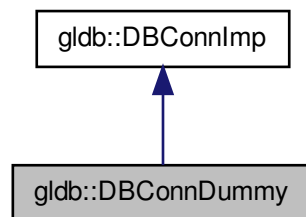
Dummy database implementation class.

```
#include <dbconn_dummy_imp.h>
```

Inheritance diagram for `gldb::DBConnDummy`:



Collaboration diagram for `gldb::DBConnDummy`:



Public Member Functions

- [DBConnDummy](#) (const std::string database, const std::string hostname, const std::string username, const std::string password)

Constructor.

- [DBConnDummy](#) (const [DBConnDummy](#) &)
- virtual [~DBConnDummy](#) ()
- [DBConnDummy](#) & [operator=](#) (const [DBConnDummy](#) &)
- virtual void [query](#) (const std::string &sql_query)

Runs an SQL query.

- [Table select](#) (const std::string &[query](#))

Fakes running of an SQL SELECT query.

9.12.1 Detailed Description

Dummy database implementation class.

9.12.2 Constructor & Destructor Documentation

9.12.2.1 `DBConnDummy::DBConnDummy (const std::string database, const std::string hostname, const std::string username, const std::string password)`

Constructor.

Parameters

<i>database</i>	The name of the Dummy database.
<i>hostname</i>	The hostname of the server.
<i>username</i>	The username to log into the database.
<i>password</i>	The password to log into the database.

9.12.2.2 `glldb::DBConnDummy::DBConnDummy (const DBConnDummy &)`

Deleted copy constructor

9.12.2.3 `DBConnDummy::~~DBConnDummy ()` [virtual]

Destructor

9.12.3 Member Function Documentation

9.12.3.1 `DBConnDummy& glldb::DBConnDummy::operator= (const DBConnDummy &)`

Deleted assignment operator

9.12.3.2 `void DBConnDummy::query (const std::string & sql_query)` [virtual]

Runs an SQL query.

Parameters

<i>sql_query</i>	The query.
------------------	------------

Exceptions

<i>DBConnCouldNotQuery</i>	If could not successfully execute query.
--	--

Implements [glldb::DBConnImp](#).

9.12.3.3 `Table DBConnDummy::select (const std::string & query)` [virtual]

Fakes running of an SQL SELECT query.

Parameters

<i>query</i>	Any query.
--------------	------------

Returns

A [Table](#) object containing dummy results.

Implements [glldb::DBConnImp](#).

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

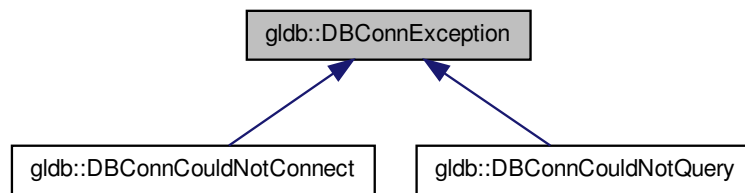
- [lib/database_imp/dummy/dbconn_dummy_imp.h](#)
- [lib/database_imp/dummy/dbconn_dummy_imp.cpp](#)

9.13 gldb::DBConnException Class Reference

Base database connection exception class.

```
#include <dbconn.h>
```

Inheritance diagram for gldb::DBConnException:



Public Member Functions

- [DBConnException](#) (const std::string &msg)
Constructor.

9.13.1 Detailed Description

Base database connection exception class.

9.13.2 Constructor & Destructor Documentation

9.13.2.1 `gldb::DBConnException::DBConnException (const std::string & msg) [inline], [explicit]`

Constructor.

Parameters

<i>msg</i>	Database error message
------------	------------------------

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

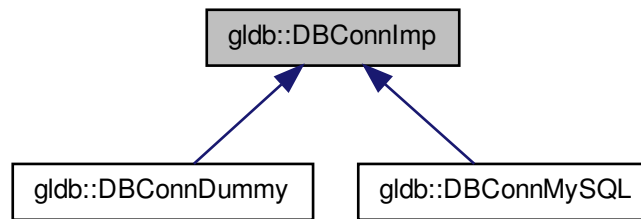
- [lib/database/dbconn.h](#)

9.14 gldb::DBConnImp Class Reference

Abstract database implementation base class.

```
#include <dbconnimp.h>
```

Inheritance diagram for glldb::DBConnImp:



Public Member Functions

- [DBConnImp](#) ()
- virtual [~DBConnImp](#) ()
- virtual void [query](#) (const std::string &sql_query)=0
Runs an SQL query.
- virtual [Table select](#) (const std::string &query)=0
Runs an SQL SELECT query.
- virtual void [begin_transaction](#) ()=0
Begins a transaction.
- virtual void [rollback_transaction](#) ()=0
Rolls back a transaction.
- virtual void [commit_transaction](#) ()=0
Commits a transaction.
- virtual unsigned long long [last_auto_increment](#) ()=0
Returns the last auto incremented value.

9.14.1 Detailed Description

Abstract database implementation base class.

9.14.2 Constructor & Destructor Documentation

9.14.2.1 glldb::DBConnImp::DBConnImp () [inline]

Constructor

9.14.2.2 virtual glldb::DBConnImp::~~DBConnImp () [inline], [virtual]

Destructor

9.14.3 Member Function Documentation

9.14.3.1 `virtual unsigned long long glldb::DBConnImp::last_auto_increment ()` [pure virtual]

Returns the last auto incremented value.

Returns

The last auto incremented value.

Implemented in [glldb::DBConnMySQL](#).

9.14.3.2 `virtual void glldb::DBConnImp::query (const std::string & sql_query)` [pure virtual]

Runs an SQL query.

Parameters

<i>sql_query</i>	The query.
------------------	------------

Implemented in [glldb::DBConnMySQL](#), and [glldb::DBConnDummy](#).

9.14.3.3 `virtual Table glldb::DBConnImp::select (const std::string & query)` [pure virtual]

Runs an SQL SELECT query.

Parameters

<i>query</i>	The query.
--------------	------------

Returns

A [Table](#) object containing the results.

Implemented in [glldb::DBConnMySQL](#), and [glldb::DBConnDummy](#).

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

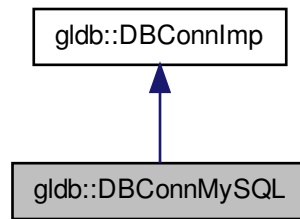
- [lib/database/dbconnimp.h](#)

9.15 glldb::DBConnMySQL Class Reference

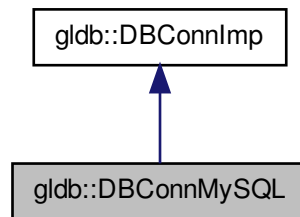
MySQL database implementation class.

```
#include <dbconn_mysql_imp.h>
```

Inheritance diagram for glldb::DBConnMySQL:



Collaboration diagram for glldb::DBConnMySQL:



Public Member Functions

- [DBConnMySQL](#) (const std::string &database, const std::string &hostname, const std::string &username, const std::string &password)
Constructor.
- [DBConnMySQL](#) (const [DBConnMySQL](#) &)
- [DBConnMySQL](#) (const [DBConnMySQL](#) &&)
- virtual [~DBConnMySQL](#) ()
- [DBConnMySQL](#) & [operator=](#) (const [DBConnMySQL](#) &)
- [DBConnMySQL](#) & [operator=](#) (const [DBConnMySQL](#) &&)
- virtual void [query](#) (const std::string &sql_query)
Runs an SQL query.
- virtual [Table select](#) (const std::string &sql_query)
Runs an SQL SELECT query.
- virtual void [begin_transaction](#) ()
Begins a transaction.
- virtual void [rollback_transaction](#) ()
Rolls back a transaction.
- virtual void [commit_transaction](#) ()
Commits a transaction.
- virtual unsigned long long [last_auto_increment](#) ()
Returns the last auto incremented value.

Private Attributes

- MySQL * [m_conn](#)

Static Private Attributes

- static std::mutex [mtx](#)

9.15.1 Detailed Description

MySQL database implementation class.

9.15.2 Constructor & Destructor Documentation

9.15.2.1 DBConnMySQL::DBConnMySQL (const std::string & *database*, const std::string & *hostname*, const std::string & *username*, const std::string & *password*)

Constructor.

Parameters

<i>database</i>	The name of the MySQL database.
<i>hostname</i>	The hostname of the server.
<i>username</i>	The username to log into the database.
<i>password</i>	The password to log into the database.

Exceptions

DBConnCouldNotConnect	If could not connect to database.
---------------------------------------	-----------------------------------

9.15.2.2 glldb::DBConnMySQL::DBConnMySQL (const DBConnMySQL &)

Deleted copy constructor

9.15.2.3 glldb::DBConnMySQL::DBConnMySQL (const DBConnMySQL &&)

Delete move constructor

9.15.2.4 virtual glldb::DBConnMySQL::~~DBConnMySQL () [virtual]

Destructor

9.15.3 Member Function Documentation

9.15.3.1 virtual unsigned long long glldb::DBConnMySQL::last_auto_increment () [virtual]

Returns the last auto incremented value.

Returns

The last auto incremented value.

Implements [glldb::DBConnImp](#).

9.15.3.2 DBConnMySQL& gldb::DBConnMySQL::operator= (const DBConnMySQL &)

Deleted assignment operator

9.15.3.3 DBConnMySQL& gldb::DBConnMySQL::operator= (const DBConnMySQL &&)

Deleted move assignment operator

9.15.3.4 virtual void gldb::DBConnMySQL::query (const std::string & *sql_query*) [virtual]

Runs an SQL query.

Parameters

<i>sql_query</i>	The SQL query.
------------------	----------------

Exceptions

<i>DBConnCouldNotQuery</i>	If could not successfully execute query.
--	--

Implements [gldb::DBConnImp](#).

9.15.3.5 virtual Table gldb::DBConnMySQL::select (const std::string & *sql_query*) [virtual]

Runs an SQL SELECT query.

Parameters

<i>sql_query</i>	The SQL query.
------------------	----------------

Returns

A [Table](#) object containing the results.

Exceptions

<i>DBConnCouldNotQuery</i>	If could not successfully execute query.
--	--

Implements [gldb::DBConnImp](#).

9.15.4 Member Data Documentation

9.15.4.1 MYSQL* gldb::DBConnMySQL::m_conn [private]

The initialized MySQL handle.

9.15.4.2 std::mutex DBConnMySQL::mtx [static], [private]

Database connection mutex

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

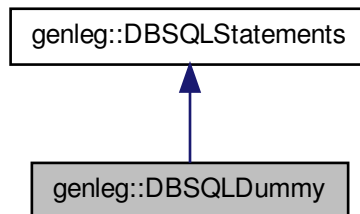
- lib/database_imp/mysql/dbconn_mysql_imp.h
- lib/database_imp/mysql/dbconn_mysql_imp.cpp

9.16 genleg::DBSQLDummy Class Reference

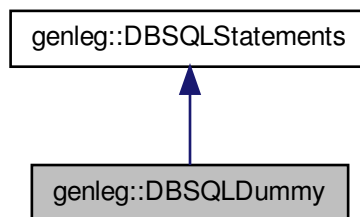
Dummy SQL statements class.

```
#include <dbsql_dummy.h>
```

Inheritance diagram for genleg::DBSQLDummy:



Collaboration diagram for genleg::DBSQLDummy:



Additional Inherited Members

9.16.1 Detailed Description

Dummy SQL statements class.

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

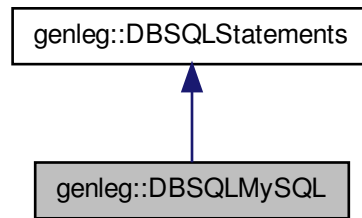
- `lib/dbsql/dbsql_dummy.h`

9.17 genleg::DBSQLMySQL Class Reference

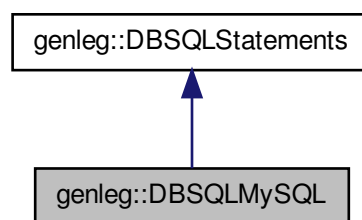
MySQL SQL statements class.

```
#include <dbsql_mysql.h>
```

Inheritance diagram for genleg::DBSQLMySQL:



Collaboration diagram for genleg::DBSQLMySQL:



Additional Inherited Members

9.17.1 Detailed Description

MySQL SQL statements class.

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

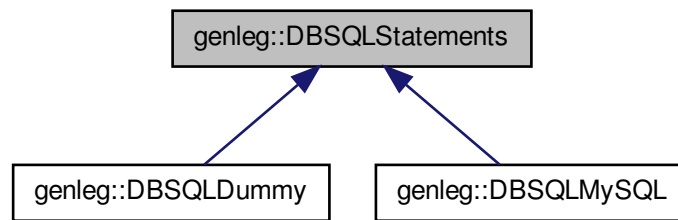
- lib/dbsql/[dbsql_mysql.h](#)

9.18 genleg::DBSQLStatements Class Reference

SQL statements class.

```
#include <dbsqlstatements.h>
```

Inheritance diagram for `genleg::DBSQLStatements`:



Public Member Functions

- [DBSQLStatements](#) ()
- virtual [~DBSQLStatements](#) ()
- virtual std::string [create_table](#) (const std::string &table_name) const
Returns a SQL statement for creating a table.
- virtual std::string [drop_table](#) (const std::string &table_name) const
Returns a SQL statement for dropping a table.
- virtual std::string [create_view](#) (const std::string &view_name) const
Returns a SQL statement for creating a view.
- virtual std::string [drop_view](#) (const std::string &view_name) const
Returns a SQL statement for dropping a view.
- virtual std::string [standing_data](#) () const
Returns a SQL statement to get the standing data.
- virtual std::string [user_by_id](#) (const std::string &user_id) const
Returns a SQL statement to select a user by ID.
- virtual std::string [user_by_username](#) (const std::string &user_name) const
Returns a SQL statement to select a user by username.
- virtual std::string [update_user](#) (const [GLUser](#) &user) const
Returns a SQL UPDATE statement to update a user.
- virtual std::string [entity_by_id](#) (const std::string &entity_id) const
Returns a SQL statement to select an entity by ID.
- virtual std::string [entity_by_name](#) (const std::string &entity_name) const
Returns a SQL statement to select an entity by short name.
- virtual std::string [account_by_name](#) (const std::string &acc_name) const
Returns a SQL statement to select a nominal account by name/number.
- virtual std::string [je_by_id](#) (const std::string &je_id) const
Returns a SQL statement to select a journal entry by ID.
- virtual std::string [jelines_by_id](#) (const std::string &je_id) const
Returns a SQL statement to select journal entry lines by ID.
- virtual std::string [post_je](#) (const unsigned int user, const unsigned int entity, const int period, const int year, const std::string &source, const std::string &memo) const
Returns a SQL INSERT statement to post a journal entry.
- virtual std::string [post_je_line](#) (const unsigned long long je, const std::string account, const std::string amount) const

- Returns a SQL INSERT query to post a journal entry line.*
- virtual std::string [grant](#) (const std::string &user_id, const std::string &perm) const
Returns a SQL statement to grant a user a permission.
- virtual std::string [revoke](#) (const std::string &user_id, const std::string &perm) const
Returns a SQL UPDATE statement to revoke a permission from a user.
- virtual std::string [get_perms](#) (const std::string &user_id) const
Returns a SQL UPDATE statement to list a user's permissions.
- virtual std::string [currenttb](#) () const
Returns a SQL statement to run the current trial balance report.
- virtual std::string [currenttb_by_entity](#) (const std::string &entity) const
Returns a SQL statement to run the current trial balance report by entity.
- std::string [listusers](#) () const
Returns a SQL statement to run the list users report.

9.18.1 Detailed Description

SQL statements class.

9.18.2 Constructor & Destructor Documentation

9.18.2.1 DBSQLStatements::DBSQLStatements ()

Constructor

9.18.2.2 DBSQLStatements::~~DBSQLStatements () [virtual]

Destructor

9.18.3 Member Function Documentation

9.18.3.1 std::string DBSQLStatements::account_by_name (const std::string &acc_name) const [virtual]

Returns a SQL statement to select a nominal account by name/number.

Parameters

<i>acc_name</i>	The account name/number.
-----------------	--------------------------

Returns

The SQL statement.

9.18.3.2 std::string DBSQLStatements::create_table (const std::string &table_name) const [virtual]

Returns a SQL statement for creating a table.

Parameters

<i>table_name</i>	The table to create.
-------------------	----------------------

Returns

The SQL statement to create the table.

9.18.3.3 `std::string DBSQLStatements::create_view (const std::string & view_name) const` `[virtual]`

Returns a SQL statement for creating a view.

Parameters

<i>view_name</i>	The view to create.
------------------	---------------------

Returns

The SQL statement to create the view.

9.18.3.4 `std::string DBSQLStatements::currenttb () const` `[virtual]`

Returns a SQL statement to run the current trial balance report.

Returns

The SQL statement.

9.18.3.5 `std::string DBSQLStatements::currenttb_by_entity (const std::string & entity) const` `[virtual]`

Returns a SQL statement to run the current trial balance report by entity.

Parameters

<i>entity</i>	The entity number for which to run the report.
---------------	--

Returns

The SQL statement.

9.18.3.6 `std::string DBSQLStatements::drop_table (const std::string & table_name) const` `[virtual]`

Returns a SQL statement for dropping a table.

Parameters

<i>table_name</i>	The table to drop.
-------------------	--------------------

Returns

The SQL statement to drop the table.

9.18.3.7 `std::string DBSQLStatements::drop_view (const std::string & view_name) const` `[virtual]`

Returns a SQL statement for dropping a view.

Parameters

<i>view_name</i>	The view to drop.
------------------	-------------------

Returns

The SQL statement to drop the view.

9.18.3.8 `std::string DBSQLStatements::entity_by_id (const std::string & entity_id) const` [virtual]

Returns a SQL statement to select an entity by ID.

Parameters

<i>entity_id</i>	The entity ID.
------------------	----------------

Returns

The SQL statement.

9.18.3.9 `std::string DBSQLStatements::entity_by_name (const std::string & entity_name) const` [virtual]

Returns a SQL statement to select an entity by short name.

Parameters

<i>entity_name</i>	The entity short name.
--------------------	------------------------

Returns

The SQL statement.

9.18.3.10 `std::string DBSQLStatements::get_perms (const std::string & user_id) const` [virtual]

Returns a SQL UPDATE statement to list a user's permissions.

Parameters

<i>user_id</i>	The user ID for which to list.
----------------	--------------------------------

Returns

The SQL statement.

9.18.3.11 `std::string DBSQLStatements::grant (const std::string & user_id, const std::string & perm) const` [virtual]

Returns a SQL statement to grant a user a permission.

Attention

This function always sets the user granting the permission to user 1. This will need to be updated to support the recording of which user has granted the permission, when support for others to be able to do so is implemented.

Parameters

<i>user_id</i>	The user ID for which to grant the permission.
<i>perm</i>	A string containing the name of the permission.

Returns

The SQL statement.

9.18.3.12 `std::string DBSQLStatements::je_by_id (const std::string & je_id) const` [virtual]

Returns a SQL statement to select a journal entry by ID.

Parameters

<i>je_id</i>	The journal entry ID.
--------------	-----------------------

Returns

The SQL statement.

9.18.3.13 `std::string DBSQLStatements::jelines_by_id (const std::string & je_id) const` [virtual]

Returns a SQL statement to select journal entry lines by ID.

Parameters

<i>je_id</i>	The journal entry ID.
--------------	-----------------------

Returns

The SQL statement.

9.18.3.14 `std::string DBSQLStatements::listusers () const`

Returns a SQL statement to run the list users report.

Returns

The SQL statement.

9.18.3.15 `std::string DBSQLStatements::post_je (const unsigned int user, const unsigned int entity, const int period, const int year, const std::string & source, const std::string & memo) const` [virtual]

Returns a SQL INSERT statement to post a journal entry.

Parameters

<i>user</i>	The ID of the posting user.
<i>entity</i>	The entity ID.
<i>period</i>	The accounting period.
<i>year</i>	The accounting year.
<i>source</i>	The journal entry source.
<i>memo</i>	The memo for the journal entry.

Returns

A string containing the query.

9.18.3.16 `std::string DBSQLStatements::post_je_line (const unsigned long long je, const std::string account, const std::string amount) const` [virtual]

Returns a SQL INSERT query to post a journal entry line.

Parameters

<i>je</i>	The journal entry ID.
<i>account</i>	The account to which to post.
<i>amount</i>	The amount to post.

Returns

A string containing the SQL statement.

9.18.3.17 `std::string DBSQLStatements::revoke (const std::string & user_id, const std::string & perm) const` [virtual]

Returns a SQL UPDATE statement to revoke a permission from a user.

Parameters

<i>user_id</i>	The user ID from which to revoke.
<i>perm</i>	The permission to revoke.

Returns

The SQL statement.

9.18.3.18 `std::string DBSQLStatements::standing_data () const` [virtual]

Returns a SQL statement to get the standing data.

Returns

The SQL statement to get the data.

9.18.3.19 `std::string DBSQLStatements::update_user (const GLUser & user) const` [virtual]

Returns a SQL UPDATE statement to update a user.

Parameters

<i>user</i>	A user object.
-------------	----------------

Returns

The SQL statement.

9.18.3.20 `std::string DBSQLStatements::user_by_id (const std::string & user_id) const` [virtual]

Returns a SQL statement to select a user by ID.

Parameters

<i>user_id</i>	The user_id
----------------	-------------

Returns

The SQL statement.

9.18.3.21 `std::string DBSQLStatements::user_by_username (const std::string & user_name) const` [virtual]

Returns a SQL statement to select a user by username.

Parameters

<i>user_name</i>	The username.
------------------	---------------

Returns

The SQL statement.

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

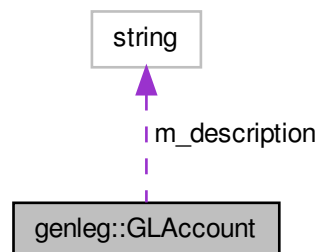
- [lib/dbsql/dbsqlstatements.h](#)
- [lib/dbsql/dbsqlstatements.cpp](#)

9.19 genleg::GLAccount Class Reference

Nominal account class.

```
#include <glaccount.h>
```

Collaboration diagram for genleg::GLAccount:



Public Member Functions

- [GLAccount](#) (const std::string number, const std::string description, const bool enabled)

Constructor.

Public Attributes

- `std::string m_description`
- `bool m_enabled`

9.19.1 Detailed Description

Nominal account class.

9.19.2 Constructor & Destructor Documentation

9.19.2.1 `genleg::GLAccount::GLAccount (const std::string number, const std::string description, const bool enabled)`
`[inline]`

Constructor.

Parameters

<i>number</i>	The account number.
<i>description</i>	The account description.
<i>enabled</i>	The enabled flag.

9.19.3 Member Data Documentation

9.19.3.1 `std::string genleg::GLAccount::m_description`

Account description

9.19.3.2 `bool genleg::GLAccount::m_enabled`

Enabled flag

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

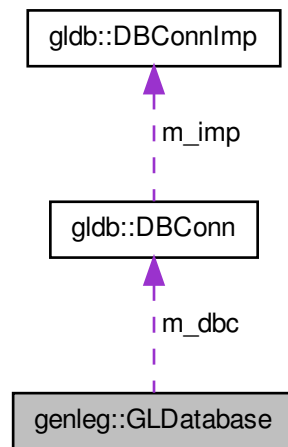
- `lib/gldb/glaccount.h`

9.20 genleg::GLDatabase Class Reference

General ledger database class.

```
#include <gldatabase.h>
```

Collaboration diagram for genleg::GLDatabase:



Public Member Functions

- [GLDatabase](#) (const std::string &database, const std::string &hostname, const std::string &username, const std::string &password)
Constructor.
- [~GLDatabase](#) ()
- void [create_structure](#) ()
Creates the database structure.
- void [destroy_structure](#) ()
Destroys the database structure.
- void [load_sample_data](#) (const std::string &dir)
Loads sample data into the database.
- [GLStandingData](#) [get_standing_data](#) ()
Gets the standing data.
- [GLUser](#) [get_user_by_id](#) (const std::string &user_id)
Returns a user from an ID.
- [GLUser](#) [get_user_by_username](#) (const std::string &user_name)
Returns a user from a user name.
- void [update_user](#) (const [GLUser](#) &user)
Updates a user's details.
- void [grant](#) (const [GLUser](#) &user, const std::string &perm)
Grants a user a permission.
- void [revoke](#) (const [GLUser](#) &user, const std::string &perm)
Revokes a permission from a user.
- [GLEntity](#) [get_entity_by_id](#) (const std::string &entity_id)
Returns an entity from an ID.
- [GLEntity](#) [get_entity_by_name](#) (const std::string &entity_name)
Returns an entity from an entity short name.
- [GLAccount](#) [get_account_by_name](#) (const std::string &acc_name)

- Returns a nominal account from an account number/name.*
- [GLJournal](#) [get_je_by_id](#) (const std::string &je_id)
Returns a journal entry from an ID.
- void [post_journal](#) (const [GLJournal](#) &journal)
Posts a journal entry.
- [GLReport](#) [report](#) (const std::string &report_name, const std::string &arg="")
Runs a report.

Static Public Member Functions

- static std::string [backend](#) ()
Returns the backend database implementation.

Private Member Functions

- [GLUser](#) [create_user](#) ([gldb::Table](#) &table)
Creates a user from a query table.
- [GLEntity](#) [create_entity](#) ([gldb::Table](#) &table)
Creates an entity from a query table.
- [GLReport](#) [standing_data_report](#) ()
Returns a standing data report.
- [GLReport](#) [current_trial_balance_report](#) (const std::string &entity)
Returns a current trial balance report.
- [GLReport](#) [list_users_report](#) ()
Returns a list users report.
- [GLReport](#) [je_report](#) (const std::string &je_id)
Returns a single journal entry report.

Private Attributes

- [gldb::DBConn](#) [m_dbc](#)
- const std::shared_ptr< const [DBSQLStatements](#) > [m_sql](#)
- const std::vector< std::string > [m_tables](#)
- const std::vector< std::string > [m_views](#)

9.20.1 Detailed Description

General ledger database class.

9.20.2 Constructor & Destructor Documentation

- 9.20.2.1 [GLDatabase::GLDatabase](#) (const std::string & *database*, const std::string & *hostname*, const std::string & *username*, const std::string & *password*)

Constructor.

Parameters

<i>database</i>	Database name.
<i>hostname</i>	Hostname of database machine.
<i>username</i>	Username to log into database.
<i>password</i>	Password to log into database.

Exceptions

<i>GLDBException</i>	on error.
--------------------------------------	-----------

9.20.2.2 GLDatabase::~~GLDatabase ()

Destructor

9.20.3 Member Function Documentation

9.20.3.1 std::string GLDatabase::backend () [static]

Returns the backend database implementation.

This may be called to discover which database platform support has been compiled into the application.

Returns

A string containing the database platform name.

9.20.3.2 GLEntity GLDatabase::create_entity (glldb::Table & table) [private]

Creates an entity from a query table.

Provided because the public functions can get a entity either from an ID or a name, this function contains the common functionality.

Parameters

<i>table</i>	A table from the appropriate query.
--------------	-------------------------------------

Returns

The new entity.

9.20.3.3 void GLDatabase::create_structure ()

Creates the database structure.

Exceptions

<i>GLDBException</i>	on error.
--------------------------------------	-----------

9.20.3.4 GLUser GLDatabase::create_user (glldb::Table & table) [private]

Creates a user from a query table.

Provided because the public functions can get a user either from an ID or a name, this function contains the common functionality.

Parameters

<i>table</i>	A table from the appropriate query.
--------------	-------------------------------------

Returns

The new user.

9.20.3.5 GLReport GLDatabase::current_trial_balance_report (const std::string & *entity*) [private]

Returns a current trial balance report.

Parameters

<i>entity</i>	The entity for which to run the report, or an empty string for all entities.
---------------	--

Returns

A [GLReport](#) object with the report.

9.20.3.6 void GLDatabase::destroy_structure ()

Destroys the database structure.

Exceptions

GLDBException	on error.
-------------------------------	-----------

9.20.3.7 GLAccount GLDatabase::get_account_by_name (const std::string & *acc_name*)

Returns a nominal account from an account number/name.

Parameters

<i>acc_name</i>	The account number/name.
-----------------	--------------------------

Returns

The account.

Exceptions

GLDBException	if the user cannot be found.
-------------------------------	------------------------------

9.20.3.8 GLEntity GLDatabase::get_entity_by_id (const std::string & *entity_id*)

Returns an entity from an ID.

Parameters

<i>entity_id</i>	The entity ID.
------------------	----------------

Returns

The entity.

Exceptions

<i>GLDBException</i>	if the entity cannot be found.
--------------------------------------	--------------------------------

9.20.3.9 GLEntity GLDatabase::get_entity_by_name (const std::string & entity_name)

Returns an entity from an entity short name.

Parameters

<i>entity_name</i>	The entity short name.
--------------------	------------------------

Returns

The entity.

Exceptions

<i>GLDBException</i>	if the user cannot be found.
--------------------------------------	------------------------------

9.20.3.10 GLJournal GLDatabase::get_je_by_id (const std::string & je_id)

Returns a journal entry from an ID.

Parameters

<i>je_id</i>	The journal entry ID.
--------------	-----------------------

Returns

The entity.

9.20.3.11 GLStandingData GLDatabase::get_standing_data ()

Gets the standing data.

Returns

The standing data.

9.20.3.12 GLUser GLDatabase::get_user_by_id (const std::string & user_id)

Returns a user from an ID.

Parameters

<i>user_id</i>	The user ID.
----------------	--------------

Returns

The user.

Exceptions

<i>GLDBException</i>	if the user cannot be found.
--------------------------------------	------------------------------

9.20.3.13 GLUser GLDatabase::get_user_by_username (const std::string & *user_name*)

Returns a user from a user name.

Parameters

<i>user_name</i>	The user name.
------------------	----------------

Returns

The user.

Exceptions

<i>GLDBException</i>	if the user cannot be found.
--------------------------------------	------------------------------

9.20.3.14 void GLDatabase::grant (const GLUser & *user*, const std::string & *perm*)

Grants a user a permission.

Parameters

<i>user</i>	The user for which to grant.
<i>perm</i>	A string containing the permission to grant.

9.20.3.15 GLReport GLDatabase::je_report (const std::string & *je_id*) [private]

Returns a single journal entry report.

Returns

A [*GLReport*](#) object with the report.

9.20.3.16 GLReport GLDatabase::list_users_report () [private]

Returns a list users report.

Returns

A [*GLReport*](#) object with the report.

9.20.3.17 void GLDatabase::load_sample_data (const std::string & *dir*)

Loads sample data into the database.

Parameters

<i>dir</i>	The directory containing the sample data. Individual files in that directory should be named after the table they are intended to populate.
------------	---

Exceptions

GLDBException	on error.
-------------------------------	-----------

9.20.3.18 void GLDatabase::post_journal (const GLJournal & *journal*)

Posts a journal entry.

Parameters

<i>journal</i>	The journal entry to post.
----------------	----------------------------

9.20.3.19 GLReport GLDatabase::report (const std::string & *report_name*, const std::string & *arg* = " ")

Runs a report.

Parameters

<i>report_name</i>	The name of the report.
<i>arg</i>	An optional argument.

Returns

A report object.

9.20.3.20 void GLDatabase::revoke (const GLUser & *user*, const std::string & *perm*)

Revokes a permission from a user.

Parameters

<i>user</i>	The user for which to revoke.
<i>perm</i>	A string containing the permission to revoke.

9.20.3.21 GLReport GLDatabase::standing_data_report () [private]

Returns a standing data report.

Returns

A [GLReport](#) object with the report.

9.20.3.22 void GLDatabase::update_user (const GLUser & *user*)

Updates a user's details.

Parameters

<i>user</i>	The user object.
-------------	------------------

9.20.4 Member Data Documentation

9.20.4.1 `gldb::DBConn genleg::GLDatabase::m_dbc` `[private]`

Database connection

9.20.4.2 `const std::shared_ptr<const DBSQLStatements> genleg::GLDatabase::m_sql` `[private]`

SQL statements object

9.20.4.3 `const std::vector<std::string> genleg::GLDatabase::m_tables` `[private]`

Vector containing database table names

9.20.4.4 `const std::vector<std::string> genleg::GLDatabase::m_views` `[private]`

Vector containing database view names

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

- [lib/gldb/gldatabase.h](#)
- [lib/gldb/gldatabase.cpp](#)

9.21 genleg::GLDBException Class Reference

Base general ledger database exceptionc class.

```
#include <glexception.h>
```

Public Member Functions

- [GLDBException](#) (const std::string &msg)
Constructor.

9.21.1 Detailed Description

Base general ledger database exceptionc class.

9.21.2 Constructor & Destructor Documentation

9.21.2.1 `genleg::GLDBException::GLDBException (const std::string & msg)` `[inline],[explicit]`

Constructor.

Parameters

<i>msg</i>	Database error message
------------	------------------------

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

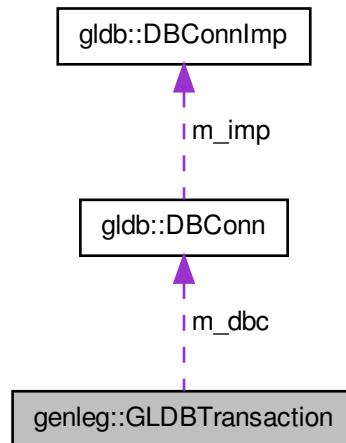
- [lib/gldb/glexception.h](#)

9.22 genleg::GLDBTransaction Class Reference

Database transaction RAI class.

```
#include <gldatabase.h>
```

Collaboration diagram for genleg::GLDBTransaction:



Public Member Functions

- [GLDBTransaction](#) ([gldb::DBConn](#) &dbc)
Constructor.
- [~GLDBTransaction](#) ()
- void [commit](#) ()
Set commit flag.

Private Attributes

- [gldb::DBConn](#) & [m_dbc](#)
- bool [m_commit](#)

9.22.1 Detailed Description

Database transaction RAI class.

9.22.2 Constructor & Destructor Documentation

9.22.2.1 genleg::GLDBTransaction::GLDBTransaction ([gldb::DBConn](#) & *dbc*) `[inline]`

Constructor.

Parameters

<i>dbc</i>	Database connection.
------------	----------------------

9.22.2.2 `genleg::GLDBTransaction::~~GLDBTransaction ()` `[inline]`

Destructor

9.22.3 Member Data Documentation

9.22.3.1 `bool genleg::GLDBTransaction::m_commit` `[private]`

Commit flag

9.22.3.2 `gldb::DBConn& genleg::GLDBTransaction::m_dbc` `[private]`

Database connection

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

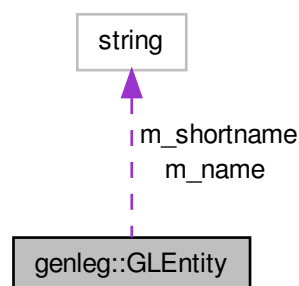
- [lib/gldb/gldatabase.h](#)

9.23 genleg::GLEntity Class Reference

General ledger entity class.

```
#include <glentity.h>
```

Collaboration diagram for genleg::GLEntity:



Public Member Functions

- [GLEntity](#) (`const size_t id`, `const std::string &name`, `const std::string &shortcode`, `const size_t parent`, `const bool aggregate`, `const bool enabled`)

Constructor.

Public Attributes

- `std::string m_name`
- `std::string m_shortcode`
- `const size_t m_parent`
- `const bool m_aggregate`
- `bool m_enabled`

9.23.1 Detailed Description

General ledger entity class.

9.23.2 Constructor & Destructor Documentation

9.23.2.1 `genleg::GLEntity::GLEntity (const size_t id, const std::string & name, const std::string & shortcode, const size_t parent, const bool aggregate, const bool enabled)` `[inline]`

Constructor.

Parameters

<i>id</i>	Entity ID.
<i>name</i>	Entity name.
<i>shortcode</i>	Entity short name.
<i>parent</i>	Parent entity ID.
<i>aggregate</i>	Aggregate entity flag.
<i>enabled</i>	Enabled flag.

9.23.3 Member Data Documentation

9.23.3.1 `const bool genleg::GLEntity::m_aggregate`

Aggregate entity flag

9.23.3.2 `bool genleg::GLEntity::m_enabled`

Enabled flag

9.23.3.3 `std::string genleg::GLEntity::m_name`

Entity name

9.23.3.4 `const size_t genleg::GLEntity::m_parent`

Parent entity ID

9.23.3.5 `std::string genleg::GLEntity::m_shortcode`

Entity short name

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

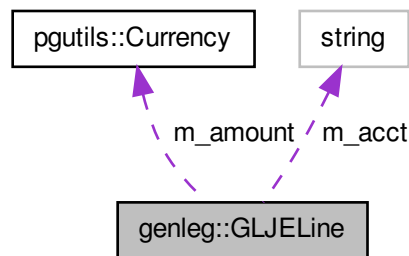
- `lib/gldb/glentity.h`

9.24 genleg::GLJELine Class Reference

Journal entry line class.

```
#include <gljournal.h>
```

Collaboration diagram for genleg::GLJELine:



Public Member Functions

- [GLJELine](#) (const std::string [account](#), const [pgutils::Currency](#) &[amount](#))
Constructor.
- std::string [account](#) () const
Returns the account name/number.
- [pgutils::Currency](#) [amount](#) () const
Returns the currency amount.

Private Attributes

- std::string [m_acct](#)
- [pgutils::Currency](#) [m_amount](#)

9.24.1 Detailed Description

Journal entry line class.

9.24.2 Constructor & Destructor Documentation

9.24.2.1 `genleg::GLJELine::GLJELine (const std::string account, const pgutils::Currency & amount)` `[inline]`

Constructor.

Parameters

<i>account</i>	The account name/ID
<i>amount</i>	The currency amount

9.24.3 Member Function Documentation

9.24.3.1 `std::string genleg::GLJELine::account () const` `[inline]`

Returns the account name/number.

Returns

The account name/number.

9.24.3.2 `pgutils::Currency genleg::GLJELine::amount () const` `[inline]`

Returns the currency amount.

Returns

The currency amount.

9.24.4 Member Data Documentation

9.24.4.1 `std::string genleg::GLJELine::m_acct` `[private]`

Account number/name

9.24.4.2 `pgutils::Currency genleg::GLJELine::m_amount` `[private]`

Amount

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

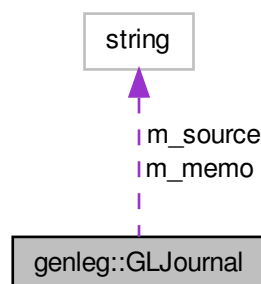
- [lib/gldb/gljournal.h](#)

9.25 `genleg::GLJournal` Class Reference

Journal entry class.

```
#include <gljournal.h>
```

Collaboration diagram for `genleg::GLJournal`:



Public Member Functions

- [GLJournal](#) (const unsigned long entity, const int period, const int year, const std::string &source, const std::string &memo, const size_t id=0, const size_t user=0)
Constructor.
- iterator [begin](#) ()
Returns an iterator to the first line.
- iterator [end](#) ()
Returns an iterator to one past the last line.
- const_iterator [begin](#) () const
Returns a const iterator to the first line.
- const_iterator [end](#) () const
Returns a const iterator to one past the last line.

Private Attributes

- unsigned long [m_entity](#)
- int [m_period](#)
- int [m_year](#)
- std::string [m_source](#)
- std::string [m_memo](#)
- size_t [m_id](#)
- size_t [m_user](#)
- std::vector< [GLJELine](#) > [m_lines](#)

9.25.1 Detailed Description

Journal entry class.

9.25.2 Constructor & Destructor Documentation

9.25.2.1 `genleg::GLJournal::GLJournal (const unsigned long entity, const int period, const int year, const std::string & source, const std::string & memo, const size_t id = 0, const size_t user = 0)` `[inline]`

Constructor.

Parameters

<i>entity</i>	The entity number.
<i>period</i>	The accounting period.
<i>year</i>	The accounting year.
<i>source</i>	The journal entry source.
<i>memo</i>	A memo for the journal entry.
<i>id</i>	The journal entry ID.
<i>user</i>	The posting user ID.

9.25.3 Member Function Documentation

9.25.3.1 `iterator genleg::GLJournal::begin ()` `[inline]`

Returns an iterator to the first line.

Returns

An iterator to the first line.

9.25.3.2 `const_iterator genleg::GLJournal::begin () const` `[inline]`

Returns a const iterator to the first line.

Returns

A const iterator to the first line.

9.25.3.3 `iterator genleg::GLJournal::end ()` `[inline]`

Returns an iterator to one past the last line.

Returns

An iterator to one past the last line.

9.25.3.4 `const_iterator genleg::GLJournal::end () const` `[inline]`

Returns a const iterator to one past the last line.

Returns

A const iterator to one past the last line.

9.25.4 Member Data Documentation

9.25.4.1 `unsigned long genleg::GLJournal::m_entity` `[private]`

The entity number for the journal entry.

9.25.4.2 `size_t genleg::GLJournal::m_id` `[private]`

The journal entry ID

9.25.4.3 `std::vector<GLJELine> genleg::GLJournal::m_lines` `[private]`

A vector of journal entry lines.

9.25.4.4 `std::string genleg::GLJournal::m_memo` `[private]`

The memo for the journal entry.

9.25.4.5 `int genleg::GLJournal::m_period` `[private]`

The accounting period.

9.25.4.6 `std::string genleg::GLJournal::m_source` `[private]`

The journal entry source.

9.25.4.7 `size_t genleg::GLJournal::m_user` `[private]`

The journal entry posting user ID

9.25.4.8 `int genleg::GLJournal::m_year` `[private]`

The accounting year.

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

- [lib/gldb/gljournal.h](#)

9.26 genleg::GLReport Class Reference

General ledger report class.

```
#include <glreport.h>
```

Public Member Functions

- [GLReport](#) (const std::string &title, const std::string &report)

Private Attributes

- const std::string [m_title](#)
- std::vector< std::pair
< std::string, std::string > > [m_headers](#)
- const std::string [m_report_text](#)

9.26.1 Detailed Description

General ledger report class.

9.26.2 Constructor & Destructor Documentation

9.26.2.1 `genleg::GLReport::GLReport (const std::string & title, const std::string & report)` `[inline]`

Constructor

9.26.3 Member Data Documentation

9.26.3.1 `std::vector<std::pair<std::string, std::string> > genleg::GLReport::m_headers` `[private]`

Report headers

9.26.3.2 `const std::string genleg::GLReport::m_report_text` [private]

The main report text

9.26.3.3 `const std::string genleg::GLReport::m_title` [private]

The report title

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

- [lib/gldb/glreport.h](#)

9.27 `genleg::GLStandingData` Class Reference

General ledger standing data class.

```
#include <glstanding.h>
```

Public Member Functions

- [GLStandingData](#) (const std::string &organization, const int period, const int year, const int num_periods)
Constructor.

Public Attributes

- int [m_period](#)
- int [m_year](#)
- int [m_num_periods](#)

9.27.1 Detailed Description

General ledger standing data class.

9.27.2 Constructor & Destructor Documentation

9.27.2.1 `genleg::GLStandingData::GLStandingData (const std::string & organization, const int period, const int year, const int num_periods)` [inline]

Constructor.

Parameters

<i>organization</i>	The overall organization.
<i>period</i>	The current accounting period.
<i>year</i>	The current accounting year.
<i>num_periods</i>	The number of accounting periods in a year.

9.27.3 Member Data Documentation

9.27.3.1 `int genleg::GLStandingData::m_num_periods`

Number of periods per year

9.27.3.2 int genleg::GLStandingData::m_period

Current period

9.27.3.3 int genleg::GLStandingData::m_year

Current year

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

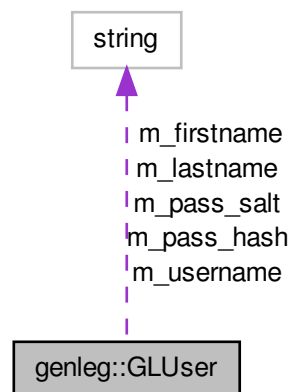
- [lib/gldb/glstanding.h](#)

9.28 genleg::GLUser Class Reference

General ledger user class.

```
#include <gluser.h>
```

Collaboration diagram for genleg::GLUser:



Public Member Functions

- [GLUser](#) (const std::string &[id](#), const std::string &[username](#), const std::string &[firstname](#), const std::string &[lastname](#), const std::string &[pass_hash](#), const std::string &[pass_salt](#), std::vector< std::string > &&perms, const bool [enabled](#))

Constructor.

- [~GLUser](#) ()
- const std::string & [id](#) () const
Returns the user ID.
- const std::string & [username](#) () const
Returns the username.
- const std::string & [firstname](#) () const
Returns the user's first name.
- const std::string & [lastname](#) () const

- Returns the user's last name.*
- const std::string & [pass_hash](#) () const
Returns the user's hashed password.
- const std::string & [pass_salt](#) () const
Returns the user's password salt.
- const std::vector< std::string > & [permissions](#) () const
Returns the permissions for a user.
- bool [enabled](#) () const
Returns the user's enabled status.
- void [set_username](#) (const std::string &new_username)
Sets a user's username.
- void [set_firstname](#) (const std::string &new_firstname)
Sets a user's first name.
- void [set_lastname](#) (const std::string &new_lastname)
Sets a user's last name.
- void [set_enabled](#) (const bool new_enabled)
Sets a user's enabled status.
- void [set_password](#) (const std::string &new_pass)
Sets a user's password hash and salt.
- bool [check_password](#) (const std::string &check_pass)
Checks a password against the user's hash.

Private Attributes

- const std::string [m_id](#)
- std::string [m_username](#)
- std::string [m_firstname](#)
- std::string [m_lastname](#)
- std::string [m_pass_hash](#)
- std::string [m_pass_salt](#)
- const std::vector< std::string > [m_perms](#)
- bool [m_enabled](#)

9.28.1 Detailed Description

General ledger user class.

9.28.2 Constructor & Destructor Documentation

- 9.28.2.1 `GLUser::GLUser (const std::string & id, const std::string & username, const std::string & firstname, const std::string & lastname, const std::string & pass_hash, const std::string & pass_salt, std::vector< std::string > && perms, const bool enabled)`

Constructor.

Parameters

<i>id</i>	User ID
<i>username</i>	Username
<i>firstname</i>	First name
<i>lastname</i>	Last name
<i>pass_hash</i>	The hashed password
<i>pass_salt</i>	The salt for the hashed password
<i>perms</i>	Vector of user permissions
<i>enabled</i>	true if user is enabled, false otherwise

9.28.2.2 GLUser::~~GLUser ()

Destructor

9.28.3 Member Function Documentation

9.28.3.1 bool GLUser::check_password (const std::string & *check_pass*)

Checks a password against the user's hash.

Parameters

<i>check_pass</i>	The password to check, must be > 8 characters.
-------------------	--

Returns

`true` is the password matches, `false` otherwise.

9.28.3.2 bool GLUser::enabled () const

Returns the user's enabled status.

Returns

The user's enabled status.

9.28.3.3 const std::string & GLUser::firstname () const

Returns the user's first name.

Returns

The user's first name.

9.28.3.4 const std::string & GLUser::id () const

Returns the user ID.

Returns

The user ID.

9.28.3.5 const std::string & GLUser::lastname () const

Returns the user's last name.

Returns

The user's last name.

9.28.3.6 `const std::string & GLUser::pass_hash () const`

Returns the user's hashed password.

Returns

The user's hashed password.

9.28.3.7 `const std::string & GLUser::pass_salt () const`

Returns the user's password salt.

Returns

The user's password salt.

9.28.3.8 `const std::vector< std::string > & GLUser::permissions () const`

Returns the permissions for a user.

Returns

A vector of strings containing the names of the permissions held by the user.

9.28.3.9 `void GLUser::set_enabled (const bool new_enabled)`

Sets a user's enabled status.

Parameters

<i>new_enabled</i>	The user's new enabled status.
--------------------	--------------------------------

9.28.3.10 `void GLUser::set_firstname (const std::string & new_firstname)`

Sets a user's first name.

Parameters

<i>new_firstname</i>	The user's new first name.
----------------------	----------------------------

9.28.3.11 `void GLUser::set_lastname (const std::string & new_lastname)`

Sets a user's last name.

Parameters

<i>new_lastname</i>	The user's new last name.
---------------------	---------------------------

9.28.3.12 `void GLUser::set_password (const std::string & new_pass)`

Sets a user's password hash and salt.

Parameters

<i>new_pass</i>	The new password, must be > 8 characters.
-----------------	---

9.28.3.13 void GLUser::set_username (const std::string & *new_username*)

Sets a user's username.

Parameters

<i>new_username</i>	The user's new username.
---------------------	--------------------------

9.28.3.14 const std::string & GLUser::username () const

Returns the username.

Returns

The username.

9.28.4 Member Data Documentation

9.28.4.1 bool genleg::GLUser::m_enabled [private]

User's enabled status

9.28.4.2 std::string genleg::GLUser::m_firstname [private]

User's first name

9.28.4.3 const std::string genleg::GLUser::m_id [private]

User ID

9.28.4.4 std::string genleg::GLUser::m_lastname [private]

User's last name

9.28.4.5 std::string genleg::GLUser::m_pass_hash [private]

User's hashed password

9.28.4.6 std::string genleg::GLUser::m_pass_salt [private]

User's password salt

9.28.4.7 const std::vector<std::string> genleg::GLUser::m_perms [private]

List of permissions

9.28.4.8 `std::string genleg::GLUser::m_username` [private]

Username

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

- [lib/gldb/gluser.h](#)
- [lib/gldb/gluser.cpp](#)
- [lib/gldb/gluser_pass.cpp](#)

9.29 gldb::MySQLResult Class Reference

MySQL result structure class.

```
#include <dbconn_mysql_result.h>
```

Public Member Functions

- [MySQLResult](#) (MYSQL *conn)
Constructor.
- [~MySQLResult](#) ()
- [MySQLResult](#) (const [MySQLResult](#) &result)
- [MySQLResult](#) ([MySQLResult](#) &&result)
- [MySQLResult](#) & operator= (const [MySQLResult](#) &result)
- [MySQLResult](#) & operator= ([MySQLResult](#) &&result)
- MYSQL_RES * [result](#) ()
Returns the MYSQL_RES pointer.
- unsigned int [num_fields](#) () const
Returns the number of fields in the result set.

Private Attributes

- MYSQL_RES * [m_result](#)
- unsigned int [m_num_fields](#)

9.29.1 Detailed Description

MySQL result structure class.

9.29.2 Constructor & Destructor Documentation

9.29.2.1 `MySQLResult::MySQLResult (MYSQL * conn)` [explicit]

Constructor.

Parameters

<i>conn</i>	MySQL connection
-------------	------------------

Exceptions

DBConnCouldNotQuery	on failure
-------------------------------------	------------

9.29.2.2 gldb::MySQLResult::~~MySQLResult ()

Destructor

9.29.2.3 gldb::MySQLResult::MySQLResult (const MySQLResult & result)

Deleted copy constructor

9.29.2.4 gldb::MySQLResult::MySQLResult (MySQLResult && result)

Deleted move constructor

9.29.3 Member Function Documentation

9.29.3.1 unsigned int gldb::MySQLResult::num_fields () const [inline]

Returns the number of fields in the result set.

Returns

The number of fields in the result set.

9.29.3.2 MySQLResult& gldb::MySQLResult::operator= (const MySQLResult & result)

Deleted copy assignment operator

9.29.3.3 MySQLResult& gldb::MySQLResult::operator= (MySQLResult && result)

Deleted move assignment operator

9.29.3.4 MYSQL_RES* gldb::MySQLResult::result () [inline]

Returns the MYSQL_RES pointer.

Returns

The MYSQL_RES pointer.

9.29.4 Member Data Documentation

9.29.4.1 unsigned int gldb::MySQLResult::m_num_fields [private]

The number of fields in the result set

9.29.4.2 MYSQL_RES* gldb::MySQLResult::m_result [private]

The MYSQL_RES pointer

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

- lib/database_imp/mysql/dbconn_mysql_result.h
- lib/database_imp/mysql/dbconn_mysql_result.cpp

9.30 pgcourses::Point Struct Reference

[Point](#) structure.

```
#include <tpgentypes.h>
```

Public Member Functions

- [Point](#) ()

Constructor.

Public Attributes

- [int](#) [x](#)
- [int](#) [y](#)

9.30.1 Detailed Description

[Point](#) structure.

9.30.2 Member Data Documentation

9.30.2.1 [int](#) [pgcourses::Point::x](#)

X coordinate

9.30.2.2 [int](#) [pgcourses::Point::y](#)

Y coordinate

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

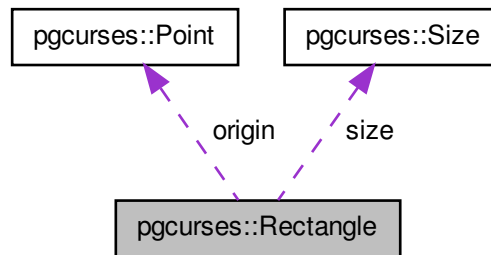
- [lib/pgcourses/tpgentypes.h](#)

9.31 pgcourses::Rectangle Struct Reference

[Rectangle](#) structure.

```
#include <tpgentypes.h>
```

Collaboration diagram for pgcurses::Rectangle:



Public Member Functions

- [Rectangle](#) ()
Constructor.

Public Attributes

- [Size](#) `size`
- [Point](#) `origin`

9.31.1 Detailed Description

[Rectangle](#) structure.

9.31.2 Member Data Documentation

9.31.2.1 Point pgcurses::Rectangle::origin

The origin

9.31.2.2 Size pgcurses::Rectangle::size

The size

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

- `lib/pgcurses/tpgentypes.h`

9.32 pgcurses::Size Struct Reference

[Size](#) structure.

```
#include <tpgentypes.h>
```

Public Member Functions

- [Size](#) ()

Constructor.

Public Attributes

- int [width](#)
- int [height](#)

9.32.1 Detailed Description

[Size](#) structure.

9.32.2 Member Data Documentation

9.32.2.1 int pgcurses::Size::height

The height

9.32.2.2 int pgcurses::Size::width

The width

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

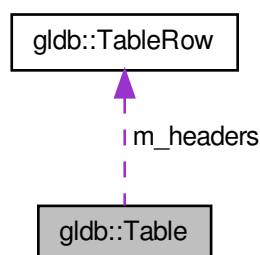
- lib/pgcurses/[tpgentypes.h](#)

9.33 glldb::Table Class Reference

Database table class.

```
#include <table.h>
```

Collaboration diagram for glldb::Table:



Public Member Functions

- [Table](#) (const [TableRow](#) &headers)
Constructor.
- [Table](#) ([TableRow](#) &&headers)
Constructor with move semantics.
- [Table](#) (const [Table](#) &table)
Copy constructor.
- [Table](#) ([Table](#) &&table)
Move constructor.
- [Table](#) & [operator=](#) (const [Table](#) &table)
Copy assignment operator.
- [Table](#) & [operator=](#) ([Table](#) &&table)
Move assignment operator.
- [~Table](#) ()
- size_t [num_fields](#) () const
Returns the number of fields in each row.
- size_t [num_records](#) () const
Returns the number of record in the table.
- iterator [begin](#) ()
Returns iterator for beginning.
- iterator [end](#) ()
Returns iterator for end plus one.
- const_iterator [begin](#) () const
Returns const iterator for beginning.
- const_iterator [end](#) () const
Returns const iterator for end plus one.
- void [set_quoted](#) (const std::vector< bool > &vec)
Sets the quote flags for the records.
- void [set_quoted](#) (std::vector< bool > &&vec)
Sets the quote flags for the records with move semantics.
- const [TableRow](#) & [get_headers](#) () const
Returns the field names.
- const [TableRow](#) & [operator\[\]](#) (const size_t idx) const
Overloaded index operator.
- void [append_record](#) (const [TableRow](#) &new_record)
Appends a record to the table.
- void [append_record](#) ([TableRow](#) &&new_record)
Appends a record to the table with move semantics.
- std::string [insert_query](#) (const std::string &table_name, const size_t idx)
Creates an SQL INSERT query from a table record.
- std::string [get_field](#) (const std::string &field_name, const size_t row_index)
Gets a field from a record by field name.

Static Public Member Functions

- static [Table](#) [create_from_file](#) (const std::string &filename, const char delim)
Creates a table from an input file.

Private Attributes

- [TableRow](#) m_headers
- std::vector< [TableRow](#) > m_records
- std::vector< bool > m_quoted

9.33.1 Detailed Description

Database table class.

9.33.2 Constructor & Destructor Documentation

9.33.2.1 Table::Table (const TableRow & headers) [explicit]

Constructor.

Parameters

<i>headers</i>	Table row containing field names.
----------------	---

9.33.2.2 Table::Table (TableRow && headers) [explicit]

Constructor with move semantics.

Parameters

<i>headers</i>	Table row containing field names.
----------------	---

9.33.2.3 Table::Table (const Table & table)

Copy constructor.

Bug 'explicit' removed from here after failure to compile at end of MySQL query function.

Parameters

<i>table</i>	Table to copy.
--------------	--------------------------------

9.33.2.4 Table::Table (Table && table)

Move constructor.

Parameters

<i>table</i>	Table to move.
--------------	--------------------------------

9.33.2.5 Table::~~Table ()

Destructor

9.33.3 Member Function Documentation

9.33.3.1 void Table::append_record (const TableRow & new_record)

Appends a record to the table.

Parameters

<i>new_record</i>	The record to append.
-------------------	-----------------------

9.33.3.2 void Table::append_record (TableRow && new_record)

Appends a record to the table with move semantics.

Parameters

<i>new_record</i>	The record to append.
-------------------	-----------------------

9.33.3.3 iterator glldb::Table::begin () [inline]

Returns iterator for beginning.

Returns

Iterator for beginning.

9.33.3.4 const_iterator glldb::Table::begin () const [inline]

Returns const iterator for beginning.

Returns

Const iterator for beginning.

9.33.3.5 Table Table::create_from_file (const std::string & filename, const char delim) [static]

Creates a table from an input file.

Parameters

<i>filename</i>	The name of the input file.
<i>delim</i>	The delimiting character.

Returns

The table.

Exceptions

<i>TableBadInputFile</i>	on badly formed input file.
<i>TableCouldNotOpenInputFile</i>	on bad filename.

9.33.3.6 iterator glldb::Table::end () [inline]

Returns iterator for end plus one.

Returns

Iterator for end plus one.

9.33.3.7 const_iterator glldb::Table::end () const [inline]

Returns const iterator for end plus one.

Returns

Const iterator for end plus one.

9.33.3.8 std::string Table::get_field (const std::string & field_name, const size_t row_index)

Gets a field from a record by field name.

Parameters

<i>field_name</i>	The name of the field.
<i>row_index</i>	The index of the row.

Returns

The contents of the field.

Exceptions

<i>TableNoSuchField</i>	if <i>field_name</i> is not a valid field name.
<i>TableNoSuchRecord</i>	if there is no record at index <i>row_index</i> .

9.33.3.9 const TableRow& glldb::Table::get_headers () const [inline]

Returns the field names.

Returns

The field names.

9.33.3.10 std::string Table::insert_query (const std::string & table_name, const size_t idx)

Creates an SQL INSERT query from a table record.

Parameters

<i>table_name</i>	The name of the table into which to INSERT.
<i>idx</i>	The index of the record.

Returns

A string containing the query.

9.33.3.11 `size_t glldb::Table::num_fields () const [inline]`

Returns the number of fields in each row.

Returns

The number of fields in each row.

9.33.3.12 `size_t glldb::Table::num_records () const [inline]`

Returns the number of record in the table.

Returns

The number of records in the table.

9.33.3.13 `Table & Table::operator= (const Table & table)`

Copy assignment operator.

Parameters

<i>table</i>	Table to copy.
--------------	--------------------------------

Returns

Reference to the assigned-to table.

9.33.3.14 `Table & Table::operator= (Table && table)`

Move assignment operator.

Parameters

<i>table</i>	Table to move.
--------------	--------------------------------

Returns

Reference to the assigned-to table.

9.33.3.15 `const TableRow & Table::operator[] (const size_t idx) const`

Overloaded index operator.

Parameters

<i>idx</i>	The zero-based index of the record.
------------	-------------------------------------

Returns

The selected record.

9.33.3.16 void Table::set_quoted (const std::vector< bool > & vec)

Sets the quote flags for the records.

Parameters

vec	A vector of bools. The size must match the size of the records.
------------	---

9.33.3.17 void Table::set_quoted (std::vector< bool > && vec)

Sets the quote flags for the records with move semantics.

Parameters

vec	A vector of bools. The size must match the size of the records.
------------	---

9.33.4 Member Data Documentation**9.33.4.1 TableRow glldb::Table::m_headers [private]**

The names of the fields

9.33.4.2 std::vector<bool> glldb::Table::m_quoted [private]

A vector to show if fields should be quoted for INSERT

9.33.4.3 std::vector<TableRow> glldb::Table::m_records [private]

A vector of the records

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

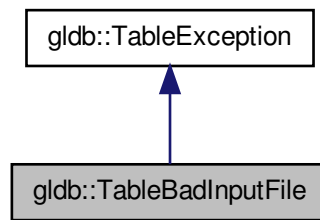
- lib/database/[table.h](#)
- lib/database/[table.cpp](#)

9.34 glldb::TableBadInputFile Class Reference

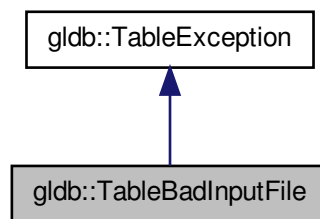
Could not connect to database exception class.

```
#include <table.h>
```

Inheritance diagram for glldb::TableBadInputFile:



Collaboration diagram for glldb::TableBadInputFile:



Public Member Functions

- [TableBadInputFile](#) (const std::string &msg)
Constructor.

9.34.1 Detailed Description

Could not connect to database exception class.

9.34.2 Constructor & Destructor Documentation

9.34.2.1 `glldb::TableBadInputFile::TableBadInputFile (const std::string & msg) [inline], [explicit]`

Constructor.

Parameters

<i>msg</i>	Database error message
------------	------------------------

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

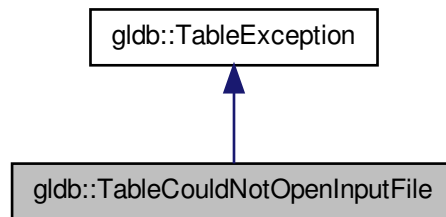
- [lib/database/table.h](#)

9.35 glldb::TableCouldNotOpenInputFile Class Reference

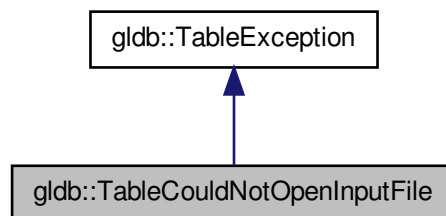
Could not connect to database exception class.

```
#include <table.h>
```

Inheritance diagram for glldb::TableCouldNotOpenInputFile:



Collaboration diagram for glldb::TableCouldNotOpenInputFile:



Public Member Functions

- [TableCouldNotOpenInputFile](#) (const std::string &msg)
Constructor.

9.35.1 Detailed Description

Could not connect to database exception class.

9.35.2 Constructor & Destructor Documentation

9.35.2.1 `glldb::TableCouldNotOpenInputFile::TableCouldNotOpenInputFile (const std::string & msg) [inline], [explicit]`

Constructor.

Parameters

<i>msg</i>	Database error message
------------	------------------------

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

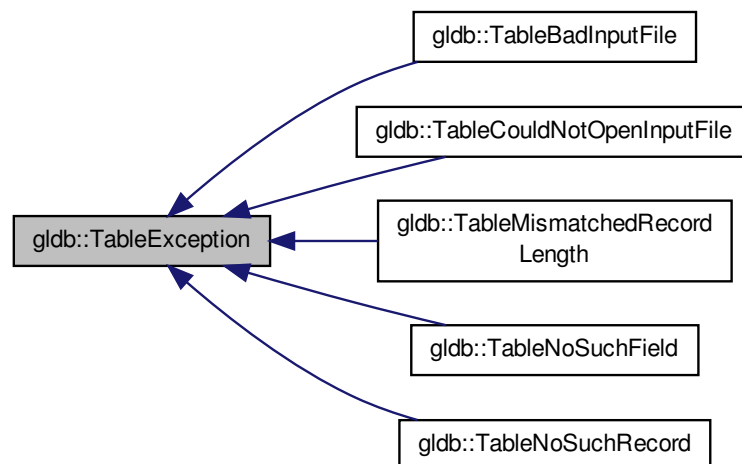
- [lib/database/table.h](#)

9.36 glldb::TableException Class Reference

Base database connection exception class.

```
#include <table.h>
```

Inheritance diagram for `glldb::TableException`:



Public Member Functions

- [TableException](#) (const std::string &msg)
Constructor.

9.36.1 Detailed Description

Base database connection exception class.

9.36.2 Constructor & Destructor Documentation

9.36.2.1 glldb::TableException::TableException (const std::string & msg) [inline],[explicit]

Constructor.

Parameters

<i>msg</i>	Database error message
------------	------------------------

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

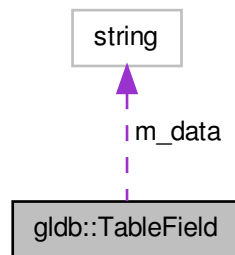
- lib/database/[table.h](#)

9.37 glldb::TableField Class Reference

Database table field class.

```
#include <tablefield.h>
```

Collaboration diagram for glldb::TableField:



Public Member Functions

- [TableField](#) (const char *data)
*Constructor accepting `const char * data`.*
- [TableField](#) (const std::string &data)
Constructor accepting `std::string data`.
- [TableField](#) (std::string &&data)
Constructor accepting `std::string data` with move semantics.
- [TableField](#) (const [TableField](#) &field)
Copy constructor.
- [TableField](#) ([TableField](#) &&field)
Move constructor.
- [~TableField](#) ()
- [size_t length](#) () const
Returns the length of the field.
- [operator std::string](#) () const
Overridden conversion operator.
- [TableField](#) & [operator=](#) (const char *data)

- Overridden assignment operator for `const char *`.*
 - [TableField](#) & [operator=](#) (`const std::string &data`)
- Overridden assignment operator for `std::string`.*
 - [TableField](#) & [operator=](#) (`std::string &&data`)
- Overridden assignment operator for `std::string` with move semantics.*
 - [TableField](#) & [operator=](#) (`const TableField &field`)
- Overridden copy assignment operator.*
 - [TableField](#) & [operator=](#) (`TableField &&field`)
- Overridden move assignment operator.*
 - `char &` [operator\[\]](#) (`const size_t idx`)
- Overridden index operator.*
 - `const char &` [operator\[\]](#) (`const size_t idx`) `const`
- Overridden index operator.*
 - [TableField](#) & [operator+=](#) (`const char c`)
- Overridden compound assignment operator.*
 - [TableField](#) & [operator+=](#) (`const std::string &data`)
- Overridden compound assignment operator.*

Private Attributes

- `std::string` [m_data](#)

Friends

- `std::ostream &` [operator<<](#) (`std::ostream &out`, `const TableField &field`)
- Overridden << operator for printing a field.*

9.37.1 Detailed Description

Database table field class.

9.37.2 Constructor & Destructor Documentation

9.37.2.1 [TableField::TableField \(const char * data \)](#) [explicit]

Constructor accepting `const char * data`.

Parameters

data	The initial contents of the field.
----------------------	------------------------------------

9.37.2.2 [TableField::TableField \(const std::string & data \)](#)

Constructor accepting `std::string data`.

Parameters

data	The initial contents of the field.
----------------------	------------------------------------

9.37.2.3 TableField::TableField (std::string && *data*)

Constructor accepting `std::string` data with move semantics.

Parameters

<i>data</i>	The initial contents of the field.
-------------	------------------------------------

9.37.2.4 TableField::TableField (const TableField & *field*)

Copy constructor.

Parameters

<i>field</i>	The field from which to copy.
--------------	-------------------------------

9.37.2.5 TableField::TableField (TableField && *field*)

Move constructor.

Parameters

<i>field</i>	The field from which to move.
--------------	-------------------------------

9.37.2.6 TableField::~~TableField ()

Destructor

9.37.3 Member Function Documentation

9.37.3.1 size_t glldb::TableField::length () const [inline]

Returns the length of the field.

Returns

The length of the field.

9.37.3.2 glldb::TableField::operator std::string () const [inline]

Overridden conversion operator.

Returns the field contents as a string.

9.37.3.3 TableField & TableField::operator+= (const char *c*)

Overridden compound assignment operator.

Parameters

<i>c</i>	The character to append to the field.
----------	---------------------------------------

Returns

A reference to the same field.

9.37.3.4 TableField & TableField::operator+= (const std::string & *data*)

Overridden compound assignment operator.

Parameters

<i>data</i>	The string to append to the field.
-------------	------------------------------------

Returns

A reference to the same field.

9.37.3.5 TableField & TableField::operator= (const char * *data*)

Overridden assignment operator for `const char *`.

Parameters

<i>data</i>	The new contents of the field.
-------------	--------------------------------

Returns

A reference to the same field.

9.37.3.6 TableField & TableField::operator= (const std::string & *data*)

Overridden assignment operator for `std::string`.

Parameters

<i>data</i>	The new contents of the field.
-------------	--------------------------------

Returns

A reference to the same field.

9.37.3.7 TableField & TableField::operator= (std::string && *data*)

Overridden assignment operator for `std::string` with move semantics.

Parameters

<i>data</i>	The new contents of the field.
-------------	--------------------------------

Returns

A reference to the same field.

9.37.3.8 TableField & TableField::operator= (const TableField & *field*)

Overridden copy assignment operator.

Parameters

<i>field</i>	The field to copy.
--------------	--------------------

Returns

A reference to the same field.

9.37.3.9 TableField & TableField::operator= (TableField && *field*)

Overridden move assignment operator.

Parameters

<i>field</i>	The field to move.
--------------	--------------------

Returns

A reference to the same field.

9.37.3.10 char& glldb::TableField::operator[] (const size_t *idx*) [inline]

Overridden index operator.

Parameters

<i>idx</i>	The desired index.
------------	--------------------

Returns

A reference to the character at the specified index.

9.37.3.11 const char& glldb::TableField::operator[] (const size_t *idx*) const [inline]

Overridden index operator.

Parameters

<i>idx</i>	The desired index.
------------	--------------------

Returns

A const reference to the character at the specified index.

9.37.4 Friends And Related Function Documentation**9.37.4.1 std::ostream& operator<< (std::ostream & *out*, const TableField & *field*) [friend]**

Overridden << operator for printing a field.

Parameters

<i>out</i>	The ostream to which to print.
<i>field</i>	A reference to the field.

Returns

A reference to `out`.

9.37.5 Member Data Documentation

9.37.5.1 `std::string glDb::TableField::m_data` `[private]`

The field contents

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

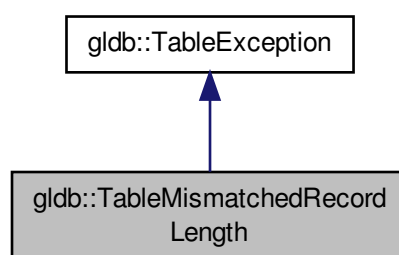
- lib/database/[tablefield.h](#)
- lib/database/[tablefield.cpp](#)

9.38 glDb::TableMismatchedRecordLength Class Reference

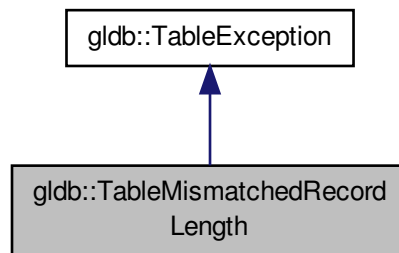
Mismatched record length exception class.

```
#include <table.h>
```

Inheritance diagram for `glDb::TableMismatchedRecordLength`:



Collaboration diagram for `gldb::TableMismatchedRecordLength`:



Public Member Functions

- [TableMismatchedRecordLength](#) (const std::string &msg)
Constructor.

9.38.1 Detailed Description

Mismatched record length exception class.

9.38.2 Constructor & Destructor Documentation

9.38.2.1 `gldb::TableMismatchedRecordLength::TableMismatchedRecordLength (const std::string & msg) [inline], [explicit]`

Constructor.

Parameters

<i>msg</i>	Database error message
------------	------------------------

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

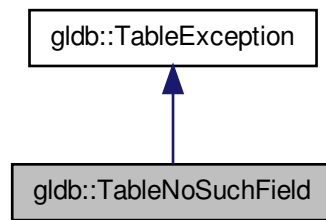
- lib/database/[table.h](#)

9.39 gldb::TableNoSuchField Class Reference

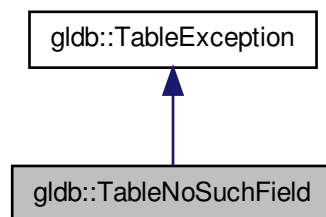
No such field exception class.

```
#include <table.h>
```

Inheritance diagram for gldb::TableNoSuchField:



Collaboration diagram for gldb::TableNoSuchField:



Public Member Functions

- [TableNoSuchField](#) (const std::string &msg)
Constructor.

9.39.1 Detailed Description

No such field exception class.

9.39.2 Constructor & Destructor Documentation

9.39.2.1 `gldb::TableNoSuchField::TableNoSuchField (const std::string & msg) [inline], [explicit]`

Constructor.

Parameters

<i>msg</i>	Database error message
------------	------------------------

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

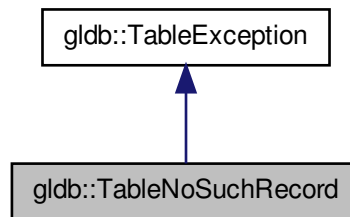
- [lib/database/table.h](#)

9.40 glDb::TableNoSuchRecord Class Reference

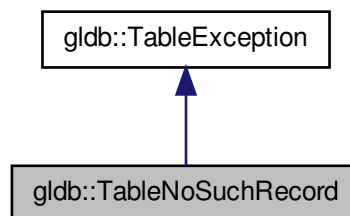
No such record exception class.

```
#include <table.h>
```

Inheritance diagram for glDb::TableNoSuchRecord:



Collaboration diagram for glDb::TableNoSuchRecord:



Public Member Functions

- [TableNoSuchRecord](#) (const std::string &msg)
Constructor.

9.40.1 Detailed Description

No such record exception class.

9.40.2 Constructor & Destructor Documentation

9.40.2.1 glldb::TableNoSuchRecord::TableNoSuchRecord (const std::string & msg) [inline],[explicit]

Constructor.

Parameters

<i>msg</i>	Database error message
------------	------------------------

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

- lib/database/[table.h](#)

9.41 glldb::TableRow Class Reference

Database table row class.

```
#include <tablerow.h>
```

Public Member Functions

- [TableRow](#) ()
- [TableRow](#) (const size_t size)
Constructor with initial number of fields.
- [TableRow](#) (const std::vector< std::string > &vec)
Constructor with string vector.
- [TableRow](#) (std::vector< std::string > &&vec)
Constructor with string vector and move semantics.
- [TableRow](#) (std::initializer_list< std::string > i)
Constructor with std::string initializer list.
- [TableRow](#) (const [TableRow](#) &row)
Copy constructor.
- [TableRow](#) ([TableRow](#) &&row)
Move constructor.
- [TableRow](#) & operator= (const [TableRow](#) &row)
Copy assignment operator.
- [TableRow](#) & operator= ([TableRow](#) &&row)
Move assignment operator.
- [~TableRow](#) ()
- size_t size () const
Returns the number of fields.
- iterator [begin](#) ()
Returns iterator for beginning.
- iterator [end](#) ()
Returns iterator for end plus one.
- const_iterator [begin](#) () const
Returns const iterator for beginning.
- const_iterator [end](#) () const
Returns const iterator for end plus one.
- [TableField](#) & operator[] (const size_t idx)
Overridden index operator.
- const [TableField](#) & operator[] (const size_t idx) const
Overridden index operator.

- void `append_field` (const char *new_field)
Appends a field to the row.
- void `append_field` (const std::string &new_field)
Appends a field to the row.
- void `append_field` (std::string &&new_field)
Appends a field to the row with move semantics.
- void `append_field` (const `TableField` &new_field)
Appends a field to the row.
- void `append_field` (`TableField` &&new_field)
Appends a field to the row with move semantics.
- void `print` (std::ostream &stream) const
Prints a row.
- std::string `record_string` (const std::vector< bool > "ed) const
Creates a comma separated string of fields.
- std::string `record_string` () const
Creates an unquoted comma separated string of fields.

Private Attributes

- std::vector< `TableField` > `m_fields`

9.41.1 Detailed Description

Database table row class.

9.41.2 Constructor & Destructor Documentation

9.41.2.1 `TableRow::TableRow ()`

Default constructor

9.41.2.2 `TableRow::TableRow (const size_t size)` `[explicit]`

Constructor with initial number of fields.

Parameters

<code>size</code>	The initial number of fields.
-------------------	-------------------------------

9.41.2.3 `TableRow::TableRow (const std::vector< std::string > & vec)` `[explicit]`

Constructor with string vector.

Parameters

<code>vec</code>	The vector.
------------------	-------------

9.41.2.4 `TableRow::TableRow (std::vector< std::string > && vec)` `[explicit]`

Constructor with string vector and move semantics.

Parameters

<i>vec</i>	The vector.
------------	-------------

9.41.2.5 TableRow::TableRow (std::initializer_list< std::string > *i*) [explicit]

Constructor with std::string initializer list.

Parameters

<i>i</i>	The initializer list.
----------	-----------------------

9.41.2.6 TableRow::TableRow (const TableRow & *row*)

Copy constructor.

Parameters

<i>row</i>	The row to copy.
------------	------------------

9.41.2.7 TableRow::TableRow (TableRow && *row*)

Move constructor.

Parameters

<i>row</i>	The row to move.
------------	------------------

9.41.2.8 TableRow::~~TableRow ()

Destructor

9.41.3 Member Function Documentation

9.41.3.1 void TableRow::append_field (const char * *new_field*)

Appends a field to the row.

Parameters

<i>new_field</i>	The contents of the new field.
------------------	--------------------------------

9.41.3.2 void TableRow::append_field (const std::string & *new_field*)

Appends a field to the row.

Parameters

<i>new_field</i>	The contents of the new field.
------------------	--------------------------------

9.41.3.3 void TableRow::append_field (std::string && *new_field*)

Appends a field to the row with move semantics.

Parameters

<i>new_field</i>	The contents of the new field.
------------------	--------------------------------

9.41.3.4 void TableRow::append_field (const TableField & *new_field*)

Appends a field to the row.

Parameters

<i>new_field</i>	A field from which to copy.
------------------	-----------------------------

9.41.3.5 void TableRow::append_field (TableField && *new_field*)

Appends a field to the row with move semantics.

Parameters

<i>new_field</i>	A field from which to copy.
------------------	-----------------------------

9.41.3.6 iterator glldb::TableRow::begin () [inline]

Returns iterator for beginning.

Returns

Iterator for beginning.

9.41.3.7 const_iterator glldb::TableRow::begin () const [inline]

Returns const iterator for beginning.

Returns

Const iterator for beginning.

9.41.3.8 iterator glldb::TableRow::end () [inline]

Returns iterator for end plus one.

Returns

Iterator for end plus one.

9.41.3.9 const_iterator glldb::TableRow::end () const [inline]

Returns const iterator for end plus one.

Returns

Const iterator for end plus one.

9.41.3.10 TableRow & TableRow::operator= (const TableRow & row)

Copy assignment operator.

Parameters

<i>row</i>	The row to copy.
------------	------------------

Returns

A reference to the assigned-to row.

9.41.3.11 TableRow & TableRow::operator= (TableRow && row)

Move assignment operator.

Parameters

<i>row</i>	The row to move.
------------	------------------

Returns

A reference to the assigned-to row.

9.41.3.12 TableField& glldb::TableRow::operator[] (const size_t idx) [inline]

Overridden index operator.

Parameters

<i>idx</i>	The zero-based index of the field.
------------	------------------------------------

Returns

A reference to the field at the specified index.

9.41.3.13 const TableField& glldb::TableRow::operator[] (const size_t idx) const [inline]

Overridden index operator.

Parameters

<i>idx</i>	The zero-based index of the field.
------------	------------------------------------

Returns

A const reference to the field at the specified index.

9.41.3.14 void TableRow::print (std::ostream & *stream*) const

Prints a row.

Parameters

<i>stream</i>	The ostream to which to print.
---------------	--------------------------------

9.41.3.15 std::string TableRow::record_string (const std::vector< bool > & *quoted*) const

Creates a comma separated string of fields.

Parameters

<i>quoted</i>	A vector of <code>bool</code> , for each field <code>true</code> means that field will be enclosed in single quotes in the comma separated string, <code>false</code> means it will not be.
---------------	---

Returns

The comma separated string.

9.41.3.16 std::string TableRow::record_string () const

Creates an unquoted comma separated string of fields.

Returns

The unquoted comma separated string.

9.41.3.17 size_t glldb::TableRow::size () const [inline]

Returns the number of fields.

Returns

The number of fields.

9.41.4 Member Data Documentation

9.41.4.1 std::vector<TableField> glldb::TableRow::m_fields [private]

A vector of fields

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

- lib/database/[tablerow.h](#)
- lib/database/[tablerow.cpp](#)

9.42 pgcurses::TermProgram Class Reference

Main terminal program class.

```
#include <termprogram.h>
```

Public Member Functions

- [TermProgram](#) ()
- [~TermProgram](#) ()
- [TermProgram](#) (const [TermProgram](#) &tp)
- [TermProgram](#) ([TermProgram](#) &&tp)
- [TermProgram](#) & [operator=](#) (const [TermProgram](#) &tp)
- [TermProgram](#) & [operator=](#) ([TermProgram](#) &&tp)
- void [sleep](#) (const unsigned int secs) const
Sleeps for a period of time.
- [Size terminal_size](#) () const
Returns the main terminal dimensions.

Private Attributes

- std::unique_ptr< [TermProgramImp](#) > m_imp

9.42.1 Detailed Description

Main terminal program class.

9.42.2 Constructor & Destructor Documentation

9.42.2.1 [TermProgram::TermProgram](#) ()

Constructor

9.42.2.2 [pgcourses::TermProgram::~~TermProgram](#) ()

Destructor

9.42.2.3 [pgcourses::TermProgram::TermProgram](#) (const [TermProgram](#) & tp)

Deleted copy constructor

9.42.2.4 [pgcourses::TermProgram::TermProgram](#) ([TermProgram](#) && tp)

Deleted move constructor

9.42.3 Member Function Documentation

9.42.3.1 [TermProgram& pgcourses::TermProgram::operator=](#) (const [TermProgram](#) & tp)

Deleted copy assignment operator

9.42.3.2 [TermProgram& pgcourses::TermProgram::operator=](#) ([TermProgram](#) && tp)

Deleted move assignment operator

9.42.3.3 `void pgcurses::TermProgram::sleep (const unsigned int secs) const`

Sleeps for a period of time.

Parameters

<code>secs</code>	The number of seconds to sleep.
-------------------	---------------------------------

9.42.3.4 `Size pgcurses::TermProgram::terminal_size () const`

Returns the main terminal dimensions.

Returns

The main terminal dimensions.

9.42.4 Member Data Documentation

9.42.4.1 `std::unique_ptr<TermProgramImp> pgcurses::TermProgram::m_imp` `[private]`

Pointer to implementation

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

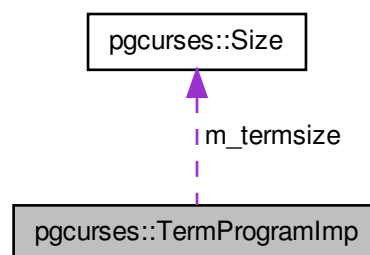
- [lib/pgcurses/termprogram.h](#)
- [lib/pgcurses/termprogram.cpp](#)

9.43 pgcurses::TermProgramImp Class Reference

Main terminal program implementation class.

```
#include <termprogramimp.h>
```

Collaboration diagram for `pgcurses::TermProgramImp`:



Public Member Functions

- [TermProgramImp \(\)](#)
- [~TermProgramImp \(\)](#)

- [TermProgramImp](#) (const [TermProgramImp](#) &tp)
- [TermProgramImp](#) ([TermProgramImp](#) &&tp)
- [TermProgramImp](#) & [operator=](#) (const [TermProgramImp](#) &tp)
- [TermProgramImp](#) & [operator=](#) ([TermProgramImp](#) &&tp)
- void [sleep](#) (const unsigned int secs) const

Sleep for a period of time.

- [Size terminal_size](#) () const

Returns the main terminal dimensions.

Private Attributes

- WINDOW * [m_mainwin](#)
- int [m_old_cursor](#)
- [Size m_termsize](#)

9.43.1 Detailed Description

Main terminal program implementation class.

9.43.2 Constructor & Destructor Documentation

9.43.2.1 [TermProgramImp::TermProgramImp \(\)](#)

Constructor

9.43.2.2 [pgcurses::TermProgramImp::~~TermProgramImp \(\)](#)

Destructor

9.43.2.3 [pgcurses::TermProgramImp::TermProgramImp \(const \[TermProgramImp\]\(#\) & *tp* \)](#)

Deleted copy constructor

9.43.2.4 [pgcurses::TermProgramImp::TermProgramImp \(\[TermProgramImp\]\(#\) && *tp* \)](#)

Deleted move constructor

9.43.3 Member Function Documentation

9.43.3.1 [TermProgramImp& pgcurses::TermProgramImp::operator= \(const \[TermProgramImp\]\(#\) & *tp* \)](#)

Deleted copy assignment operator

9.43.3.2 [TermProgramImp& pgcurses::TermProgramImp::operator= \(\[TermProgramImp\]\(#\) && *tp* \)](#)

Deleted move assignment operator

9.43.3.3 `void pgcurses::TermProgramImp::sleep (const unsigned int secs) const`

Sleep for a period of time.

Parameters

<code>secs</code>	The number of seconds to sleep.
-------------------	---------------------------------

9.43.3.4 `Size pgcurses::TermProgramImp::terminal_size () const`

Returns the main terminal dimensions.

Returns

The main terminal dimensions.

9.43.4 Member Data Documentation

9.43.4.1 `WINDOW* pgcurses::TermProgramImp::m_mainwin` `[private]`

Pointer to main window

9.43.4.2 `int pgcurses::TermProgramImp::m_old_cursor` `[private]`

Value of old cursor

9.43.4.3 `Size pgcurses::TermProgramImp::m_termsize` `[private]`

Terminal size

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

- [lib/pgcurses/termprogramimp.h](#)
- [lib/pgcurses/termprogramimp.cpp](#)

9.44 pgcurses::TPEXception Class Reference

Base terminal program exception class.

```
#include <tpexception.h>
```

Public Member Functions

- [TPEXception](#) (const std::string &msg)
Constructor.

9.44.1 Detailed Description

Base terminal program exception class.

9.44.2 Constructor & Destructor Documentation

9.44.2.1 pgcurses::TPException::TPException (const std::string & msg) [inline]

Constructor.

Parameters

<i>msg</i>	Error message
------------	---------------

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

- [lib/pgcurses/tpexception.h](#)

9.45 pgcurses::TPWindow Class Reference

Generic window class.

```
#include <tpwindow.h>
```

Public Member Functions

- [TPWindow](#) (const [Point](#) origin, const [Size](#) sz)
Constructor.
- [TPWindow](#) (const [Rectangle](#) rect)
Constructor.
- virtual [~TPWindow](#) ()
- [TPWindow](#) (const [TPWindow](#) &tp)
- [TPWindow](#) ([TPWindow](#) &&tp)
- [TPWindow](#) & operator= (const [TPWindow](#) &tp)
- [TPWindow](#) & operator= ([TPWindow](#) &&tp)
- void [write_char](#) (const char ch)
Writes a character to the current position.
- void [write_char](#) (const char ch, const [Point](#) pt)
Writes a character to a specified position.
- void [write_str](#) (const std::string &s)
Writes a string to the current position.
- void [write_str](#) (const std::string &s, const [Point](#) pt)
Writes a string to a specified position.
- void [draw](#) ()
Draws the window.
- void [redraw](#) ()
Forces a redraw of the window.

Private Attributes

- std::unique_ptr< [TPWindowImp](#) > [m_imp](#)

9.45.1 Detailed Description

Generic window class.

9.45.2 Constructor & Destructor Documentation

9.45.2.1 TPWindow::TPWindow (const Point *origin*, const Size *sz*)

Constructor.

Parameters

<i>origin</i>	The new window's origin.
<i>sz</i>	The new window's size.

Exceptions

<i>TPEException</i>	on failure.
-------------------------------------	-------------

9.45.2.2 pgcurses::TPWindow::TPWindow (const Rectangle *rect*)

Constructor.

Parameters

<i>rect</i>	The new window's rectangle.
-------------	-----------------------------

Exceptions

<i>TPEException</i>	on failure.
-------------------------------------	-------------

9.45.2.3 virtual pgcurses::TPWindow::~~TPWindow () [virtual]

Destructor

9.45.2.4 pgcurses::TPWindow::TPWindow (const TPWindow & *tp*)

Deleted copy constructor

9.45.2.5 pgcurses::TPWindow::TPWindow (TPWindow && *tp*)

Deleted move constructor

9.45.3 Member Function Documentation

9.45.3.1 TPWindow& pgcurses::TPWindow::operator= (const TPWindow & *tp*)

Deleted copy assignment operator

9.45.3.2 TPWindow& pgcurses::TPWindow::operator= (TPWindow && *tp*)

Deleted move assignment operator

9.45.3.3 void pgcurses::TPWindow::write_char (const char *ch*)

Writes a character to the current position.

Parameters

<i>ch</i>	The character to write.
-----------	-------------------------

9.45.3.4 void pgcurses::TPWindow::write_char (const char *ch*, const Point *pt*)

Writes a character to a specified position.

Parameters

<i>ch</i>	The character to write.
<i>pt</i>	The point at which to write.

9.45.3.5 void pgcurses::TPWindow::write_str (const std::string & *s*)

Writes a string to the current position.

Parameters

<i>s</i>	The string to write.
----------	----------------------

9.45.3.6 void pgcurses::TPWindow::write_str (const std::string & *s*, const Point *pt*)

Writes a string to a specified position.

Parameters

<i>s</i>	The string to write.
<i>pt</i>	The point at which to write.

9.45.4 Member Data Documentation

9.45.4.1 std::unique_ptr<TPWindowImp> pgcurses::TPWindow::m_imp [private]

Pointer to implementation

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

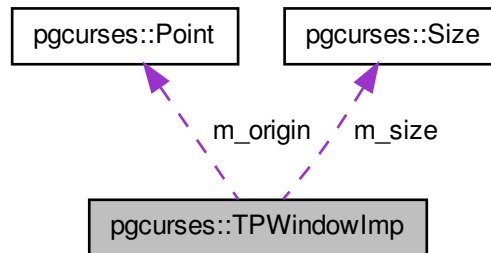
- [lib/pgcurses/tpwindow.h](#)
- [lib/pgcurses/tpwindow.cpp](#)

9.46 pgcurses::TPWindowImp Class Reference

Terminal program window implementation class.

```
#include <tpwindowimp.h>
```

Collaboration diagram for pgcurses::TPWindowImp:



Public Member Functions

- [TPWindowImp](#) (const [Point](#) origin, const [Size](#) sz)
Constructor.
- [TPWindowImp](#) (const [Rectangle](#) rect)
Constructor.
- [~TPWindowImp](#) ()
- [TPWindowImp](#) (const [TPWindowImp](#) &tp)
- [TPWindowImp](#) ([TPWindowImp](#) &&tp)
- [TPWindowImp](#) & [operator=](#) (const [TPWindowImp](#) &tp)
- [TPWindowImp](#) & [operator=](#) ([TPWindowImp](#) &&tp)
- void [write_char](#) (const char ch)
Writes a character to the current position.
- void [write_char](#) (const char ch, const [Point](#) pt)
Writes a character to a specified position.
- void [write_str](#) (const std::string &s)
Writes a string to the current position.
- void [write_str](#) (const std::string &s, const [Point](#) pt)
Writes a string to a specified position.
- void [draw](#) ()
Draws the window.
- void [redraw](#) ()
Forces a redraw of the window.

Private Attributes

- WINDOW * [m_win](#)
- [Point](#) [m_origin](#)
- [Size](#) [m_size](#)

9.46.1 Detailed Description

Terminal program window implementation class.

9.46.2 Constructor & Destructor Documentation

9.46.2.1 TPWindowImp::TPWindowImp (const Point *origin*, const Size *sz*)

Constructor.

Parameters

<i>origin</i>	The new window's origin.
<i>sz</i>	The new window's size.

Exceptions

TPEXception	on failure.
-----------------------------	-------------

9.46.2.2 pgcurses::TPWindowImp::TPWindowImp (const Rectangle *rect*)

Constructor.

Parameters

<i>rect</i>	The new window's rectangle.
-------------	-----------------------------

Exceptions

TPEXception	on failure.
-----------------------------	-------------

9.46.2.3 pgcurses::TPWindowImp::~~TPWindowImp ()

Destructor

9.46.2.4 pgcurses::TPWindowImp::TPWindowImp (const TPWindowImp & *tp*)

Deleted copy constructor

9.46.2.5 pgcurses::TPWindowImp::TPWindowImp (TPWindowImp && *tp*)

Deleted move constructor

9.46.3 Member Function Documentation

9.46.3.1 TPWindowImp& pgcurses::TPWindowImp::operator= (const TPWindowImp & *tp*)

Deleted copy assignment operator

9.46.3.2 TPWindowImp& pgcurses::TPWindowImp::operator= (TPWindowImp && *tp*)

Deleted move assignment operator

9.46.3.3 void pgcurses::TPWindowImp::write_char (const char *ch*)

Writes a character to the current position.

Parameters

<i>ch</i>	The character to write.
-----------	-------------------------

9.46.3.4 void pgcurses::TPWindowImp::write_char (const char *ch*, const Point *pt*)

Writes a character to a specified position.

Parameters

<i>ch</i>	The character to write.
<i>pt</i>	The point at which to write.

9.46.3.5 void pgcurses::TPWindowImp::write_str (const std::string & *s*)

Writes a string to the current position.

Parameters

<i>s</i>	The string to write.
----------	----------------------

9.46.3.6 void pgcurses::TPWindowImp::write_str (const std::string & *s*, const Point *pt*)

Writes a string to a specified position.

Parameters

<i>s</i>	The string to write.
<i>pt</i>	The point at which to write.

9.46.4 Member Data Documentation

9.46.4.1 Point pgcurses::TPWindowImp::m_origin [private]

The window's origin

9.46.4.2 Size pgcurses::TPWindowImp::m_size [private]

The window's size

9.46.4.3 WINDOW* pgcurses::TPWindowImp::m_win [private]

Pointer to curses WINDOW

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

- [lib/pgcurses/tpwindowimp.h](#)
- [lib/pgcurses/tpwindowimp.cpp](#)

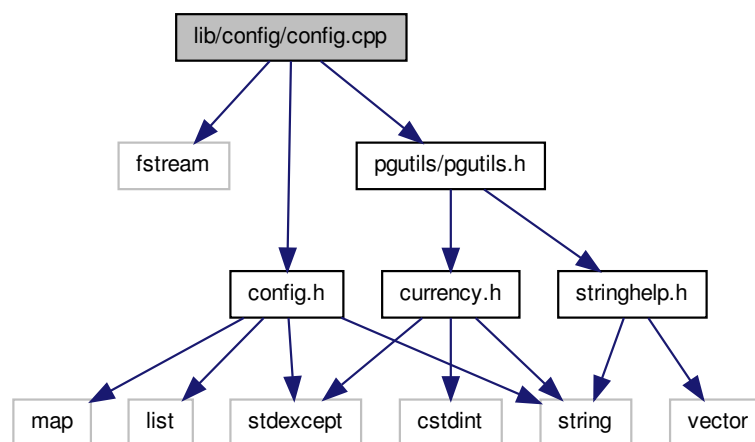
Chapter 10

File Documentation

10.1 lib/config/config.cpp File Reference

Implementation of program configurations class.

```
#include <fstream>
#include "config.h"
#include "pgutils/pgutils.h"
Include dependency graph for config.cpp:
```



10.1.1 Detailed Description

Implementation of program configurations class.

Author

Paul Griffiths

Copyright

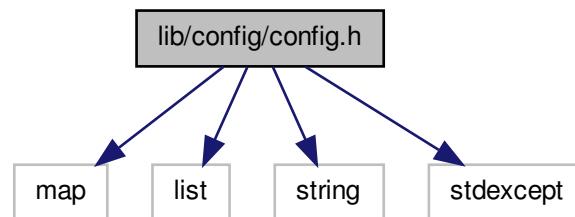
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.2 lib/config/config.h File Reference

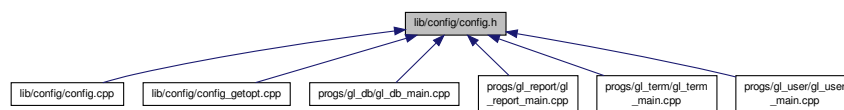
Interface to program configurations class.

```
#include <map>
#include <list>
#include <string>
#include <stdexcept>
```

Include dependency graph for config.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [genleg::ConfigException](#)
Configuration module exception base class.
- class [genleg::ConfigOptionNotSet](#)
Exception class for option not set.
- class [genleg::ConfigBadOption](#)
Exception class for bad provided option.
- class [genleg::ConfigCouldNotOpenFile](#)
Exception class for when conf file cannot be opened.
- class [genleg::ConfigBadConfigFile](#)
Exception class for badly formed configuration file.
- class [genleg::Config](#)
Configuration options class.

Enumerations

- enum [genleg::Argument](#)
Enumeration class for option argument specifications.

10.2.1 Detailed Description

Interface to program configurations class.

Author

Paul Griffiths

Copyright

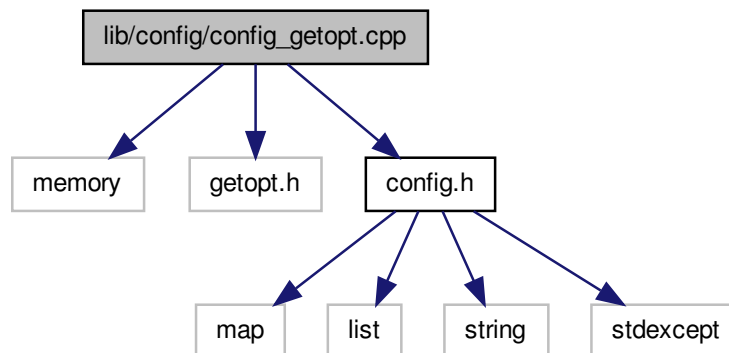
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.3 lib/config/config_getopt.cpp File Reference

Implementation of command line functionality.

```
#include <memory>
#include <getopt.h>
#include "config.h"
```

Include dependency graph for config_getopt.cpp:



Macros

- `#define _XOPEN_SOURCE 600`

10.3.1 Detailed Description

Implementation of command line functionality. Included in separate file to isolate usage of non-standard getopt library.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.3.2 Macro Definition Documentation

10.3.2.1 `#define _XOPEN_SOURCE 600`

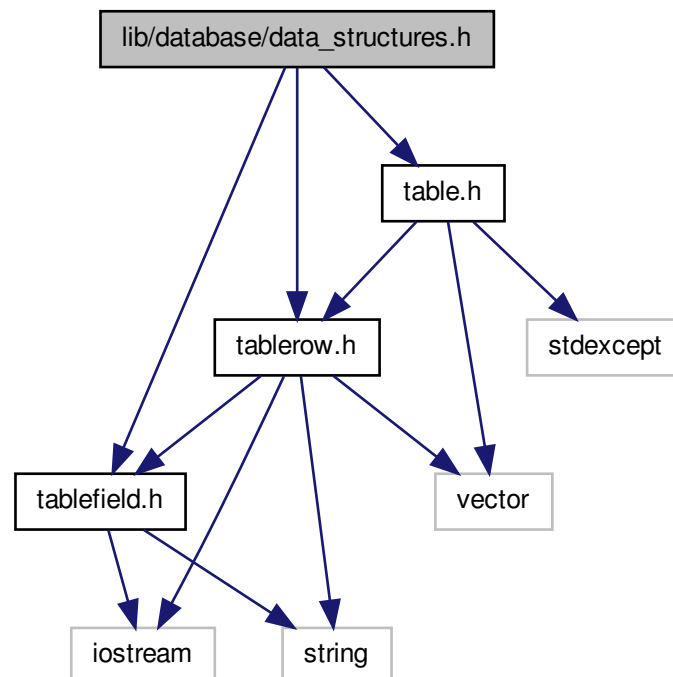
UNIX feature test macro for getopt library

10.4 `lib/database/data_structures.h` File Reference

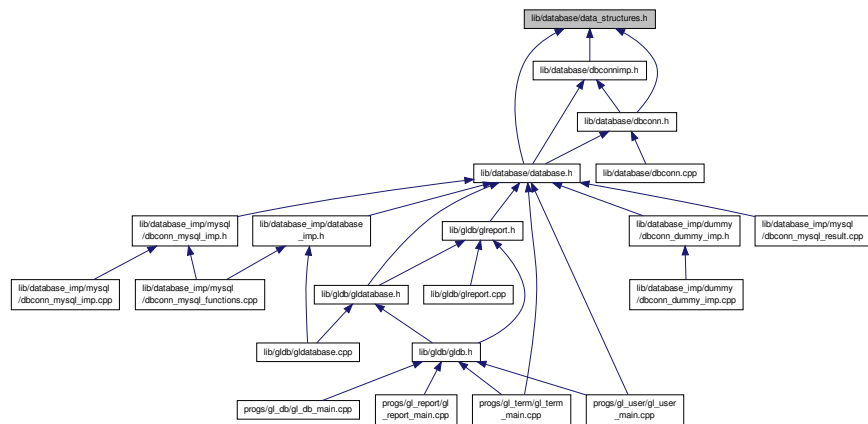
Main interface to database data structures.

```
#include "tablefield.h"  
#include "tablerow.h"  
#include "table.h"
```

Include dependency graph for `data_structures.h`:



This graph shows which files directly or indirectly include this file:



10.4.1 Detailed Description

Main interface to database data structures.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.5 lib/database/database.h File Reference

User interface to database functionality.

```
#include "data_structures.h"
#include "dbconnimp.h"
#include "dbconn.h"
```


Author

Paul Griffiths

Copyright

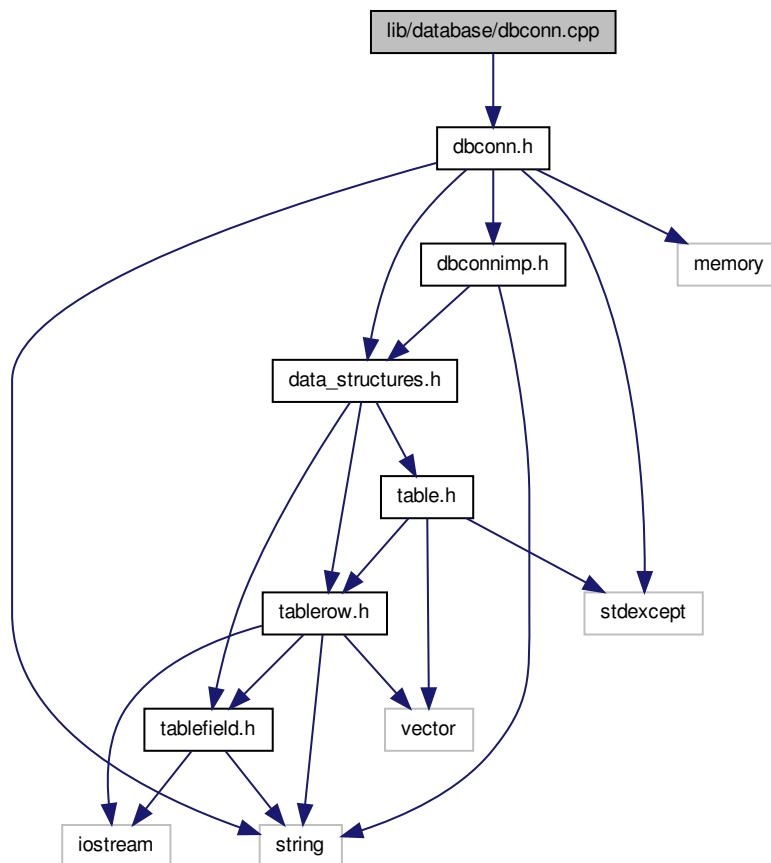
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.6 lib/database/dbconn.cpp File Reference

Implementation of database connection class.

```
#include "dbconn.h"
```

Include dependency graph for dbconn.cpp:



10.6.1 Detailed Description

Implementation of database connection class.

Author

Paul Griffiths

Copyright

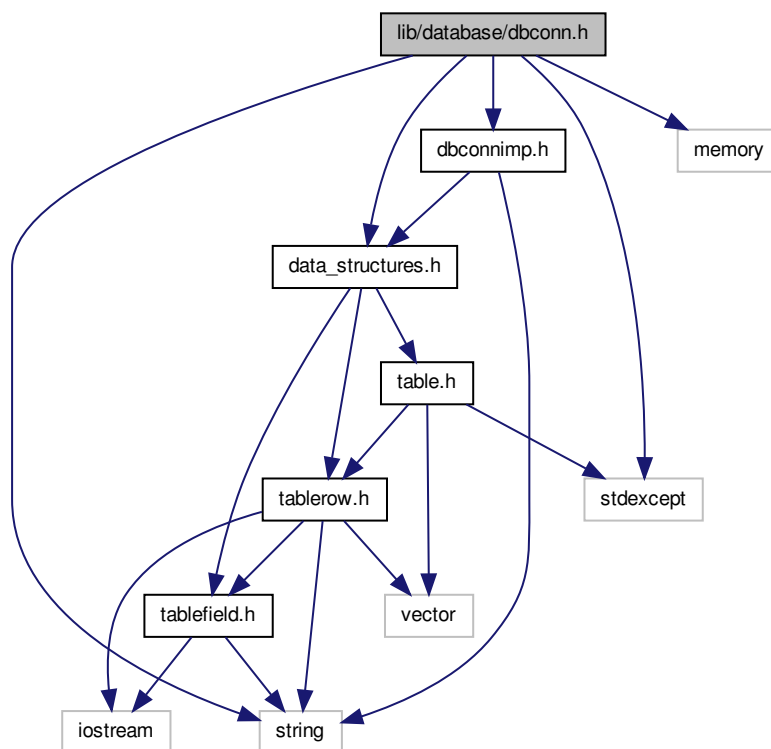
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.7 lib/database/dbconn.h File Reference

Interface to database connection base class.

```
#include <string>
#include <memory>
#include <stdexcept>
#include "data_structures.h"
#include "dbconnimp.h"
```

Include dependency graph for dbconn.h:



[illegible]

- class `gldb::DBConnException`
Base database connection exception class.
- class `gldb::DBConnCouldNotConnect`
Could not connect to database exception class.
- class `gldb::DBConnCouldNotQuery`
Could not execute database query exception class.
- class `gldb::DBConn`
Database connection class.

Interface to database connection base class.

Paul Griffiths

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

Interface to abstract database implementation base class.

```
#include <string>
#include "data_structures.h"
```


Abstract database implementation base class.

10.8.1 Detailed Description

Interface to abstract database implementation base class.

Author

Paul Griffiths

Copyright

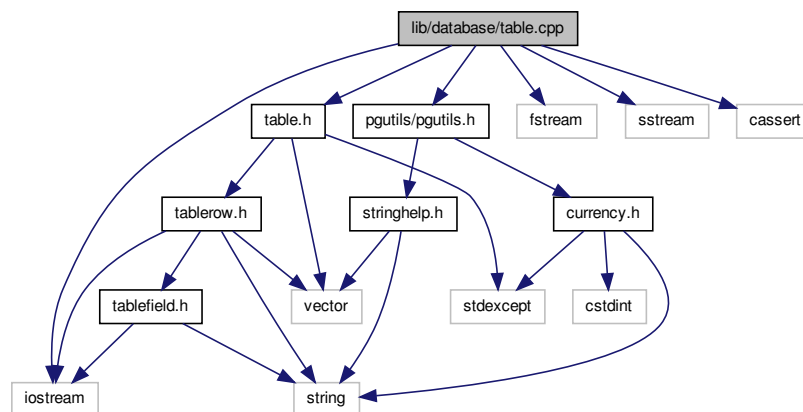
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.9 lib/database/table.cpp File Reference

Implementation of database table data structure.

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <cassert>
#include "table.h"
#include "pgutils/pgutils.h"
```

Include dependency graph for table.cpp:



10.9.1 Detailed Description

Implementation of database table data structure.

Author

Paul Griffiths

Copyright

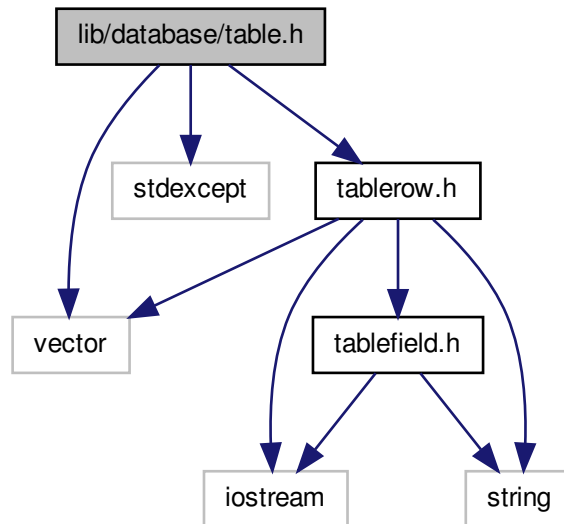
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.10 lib/database/table.h File Reference

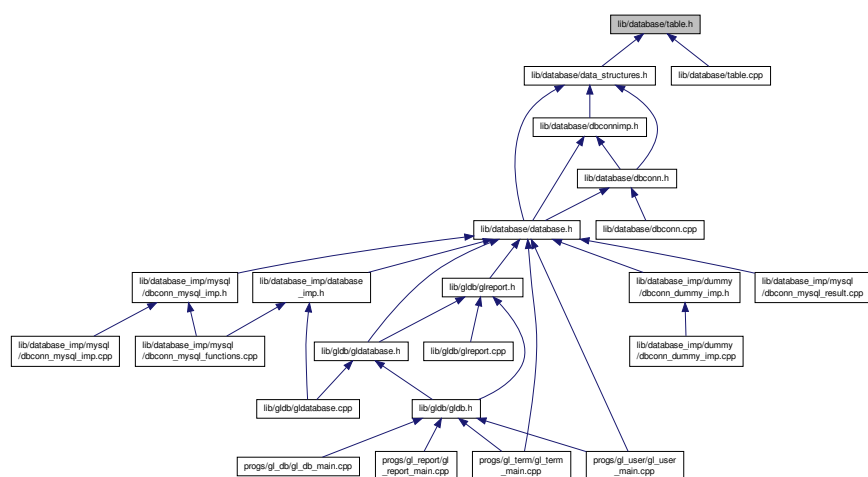
Interface to database table data structure.

```
#include <vector>
#include <stdexcept>
#include "tablerow.h"
```

Include dependency graph for table.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gldb::TableException](#)

- class `gldb::TableNoSuchField`
No such field exception class.
- class `gldb::TableNoSuchRecord`
No such record exception class.
- class `gldb::TableMismatchedRecordLength`
Mismatched record length exception class.
- class `gldb::TableBadInputFile`
Could not connect to database exception class.
- class `gldb::TableCouldNotOpenInputFile`
Could not connect to database exception class.
- class `gldb::Table`
Database table class.

10.10.1 Detailed Description

Interface to database table data structure.

Author

Paul Griffiths

Copyright

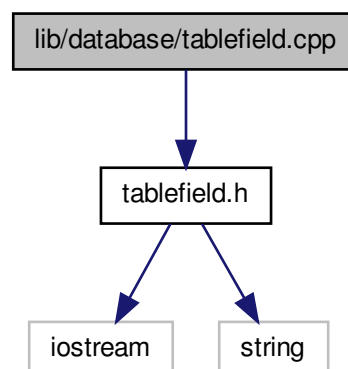
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.11 lib/database/tablefield.cpp File Reference

Implementation of database table field class.

```
#include "tablefield.h"
```

Include dependency graph for tablefield.cpp:



10.11.1 Detailed Description

Implementation of database table field class.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

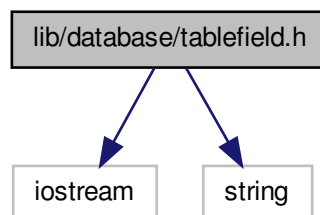
10.12 lib/database/tablefield.h File Reference

Interface to database table field class.

```
#include <iostream>
```

```
#include <string>
```

Include dependency graph for tablefield.h:



[illegible]

- class `gldb::TableField`
Database table field class.

- `std::ostream & gldb::operator<< (std::ostream &out, const TableField &field)`
Overridden << operator for printing a field.

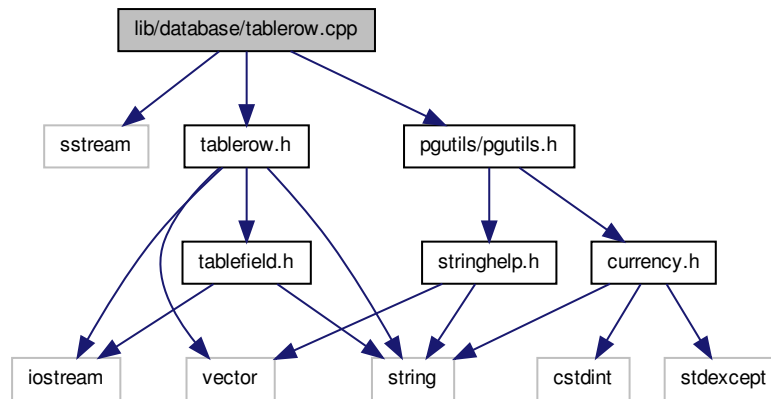
Interface to database table field class.

Author
Paul Griffiths

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

Implementation of database table row data structure.

```
#include <sstream>
#include "tablerow.h"
#include "pgutils/pgutils.h"
Include dependency graph for tablerow.cpp:
```



10.13.1 Detailed Description

Implementation of database table row data structure.

Author

Paul Griffiths

Copyright

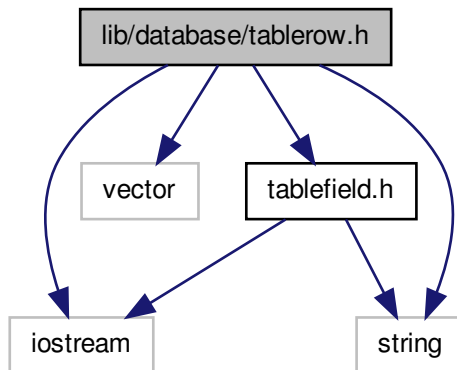
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.14 lib/database/tablerow.h File Reference

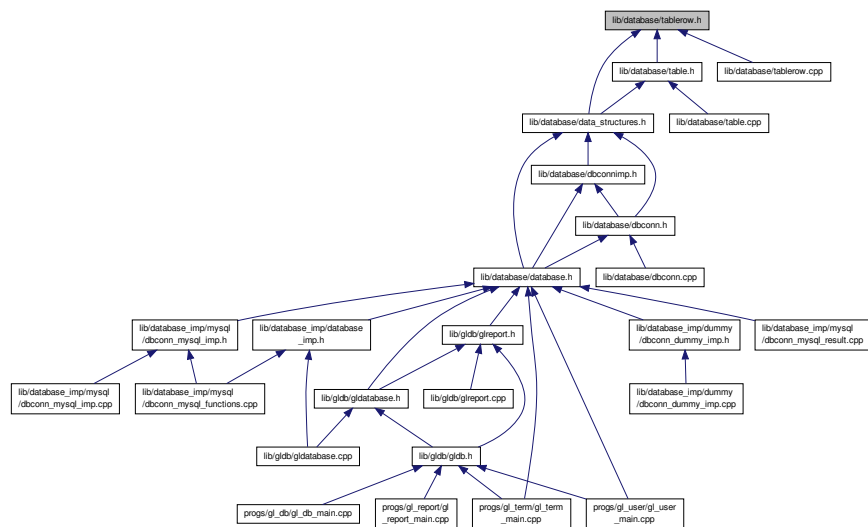
Interface to database table row data structure.

```
#include <iostream>
#include <vector>
#include <string>
#include "tablefield.h"
```


Include dependency graph for tablerow.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gldb::TableRow`
Database table row class.

10.14.1 Detailed Description

Interface to database table row data structure.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

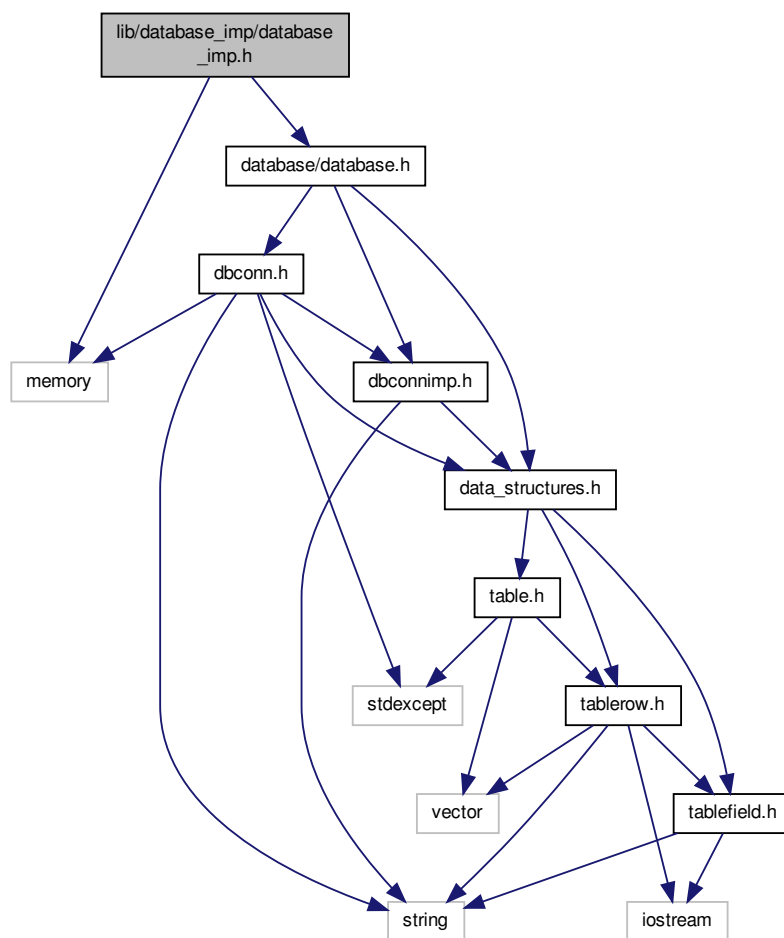
10.15 lib/database_imp/database_imp.h File Reference

Interface to database implementation factory function.

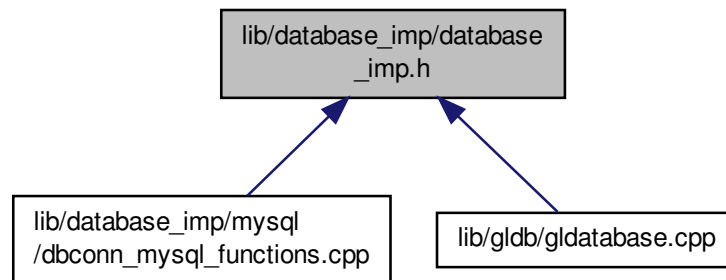
```
#include <memory>
```

```
#include "database/database.h"
```

Include dependency graph for database_imp.h:



This graph shows which files directly or indirectly include this file:



Functions

- `DBConnImp *` `gldb::get_connection` (const std::string &database, const std::string &hostname, const std::string &username, const std::string &password)

Creates and returns a pointer to a database implementation.

- std::string `gldb::get_database_type` ()

Returns the name of the compiled-in database type.

10.15.1 Detailed Description

Interface to database implementation factory function.

Author

Paul Griffiths

Copyright

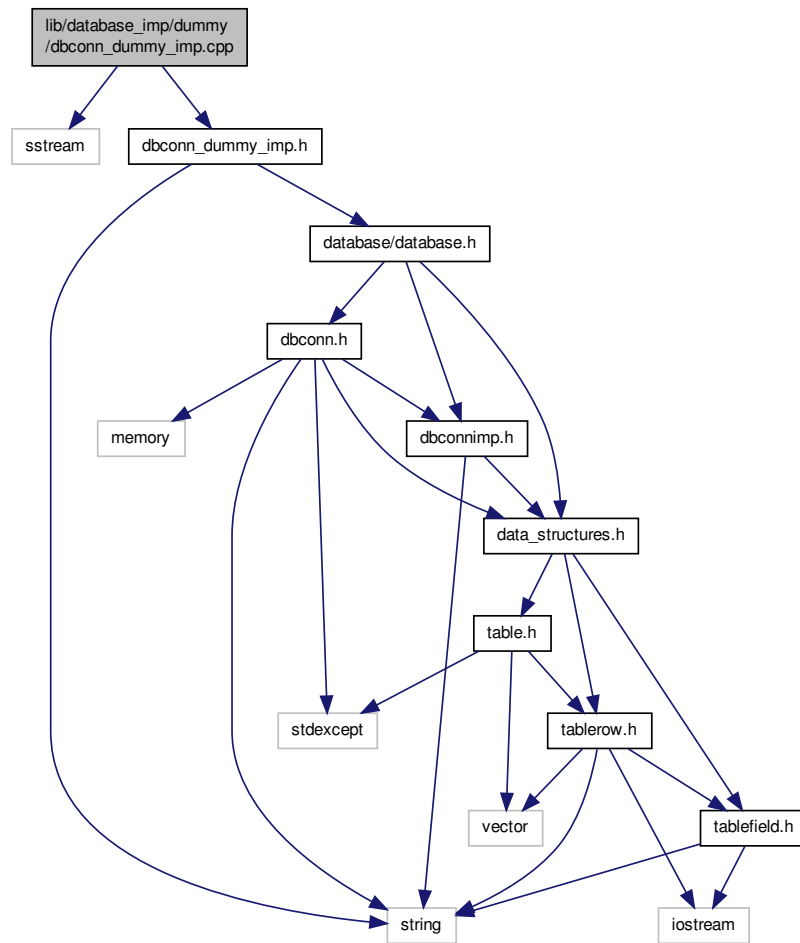
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.16 lib/database_imp/dummy/dbconn_dummy_imp.cpp File Reference

Implementation of Dummy database connection implementation class.

```
#include <sstream>
#include "dbconn_dummy_imp.h"
```

Include dependency graph for `dbconn_dummy_imp.cpp`:



10.16.1 Detailed Description

Implementation of Dummy database connection implementation class.

Author

Paul Griffiths

Copyright

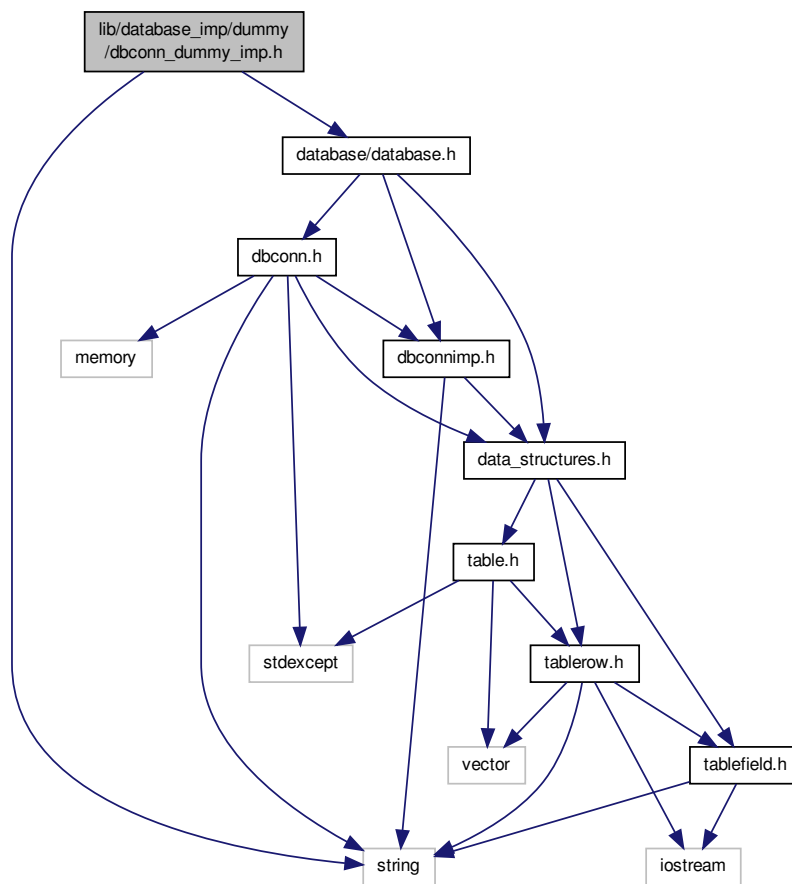
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.17 lib/database_imp/dummy/dbconn_dummy_imp.h File Reference

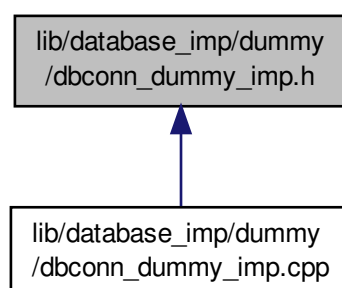
Interface to dummy database connection implementation class.

```
#include <string>
#include "database/database.h"
```

Include dependency graph for dbconn_dummy_imp.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gldb::DBConnDummy`

Dummy database implementation class.

10.17.1 Detailed Description

Interface to dummy database connection implementation class.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.18 lib/database_imp/mysql/dbconn_mysql_functions.cpp File Reference

Implementation of MySQL implementation factory function.

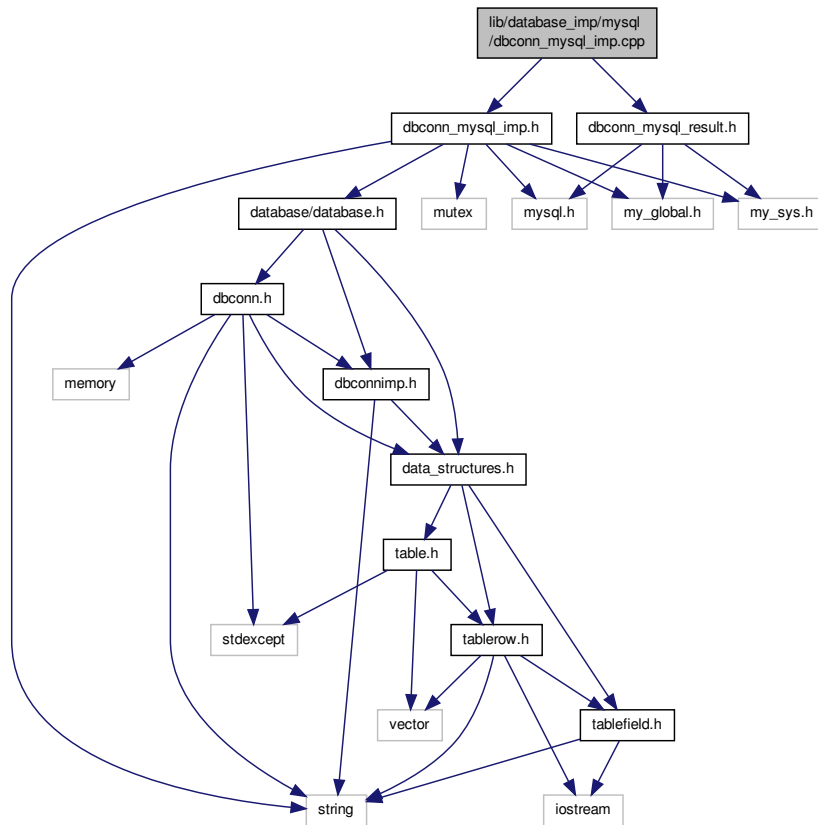
```
#include "../database_imp.h"
#include "dbconn_mysql_imp.h"
```

[illegible]

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

Implementation of MySQL database connection implementation class.

```
#include "dbconn_mysql_imp.h"
#include "dbconn_mysql_result.h"
Include dependency graph for dbconn_mysql_imp.cpp:
```



Functions

- static [TableRow get_field_names](#) ([MySQLResult](#) &result)
Gets field names from a MySQL result structure.
- static [TableRow get_row](#) ([MySQLResult](#) &result, MYSQL_ROW row)
Creates a TableRow from a MySQL result row.

10.19.1 Detailed Description

Implementation of MySQL database connection implementation class.

Author

Paul Griffiths

Copyright

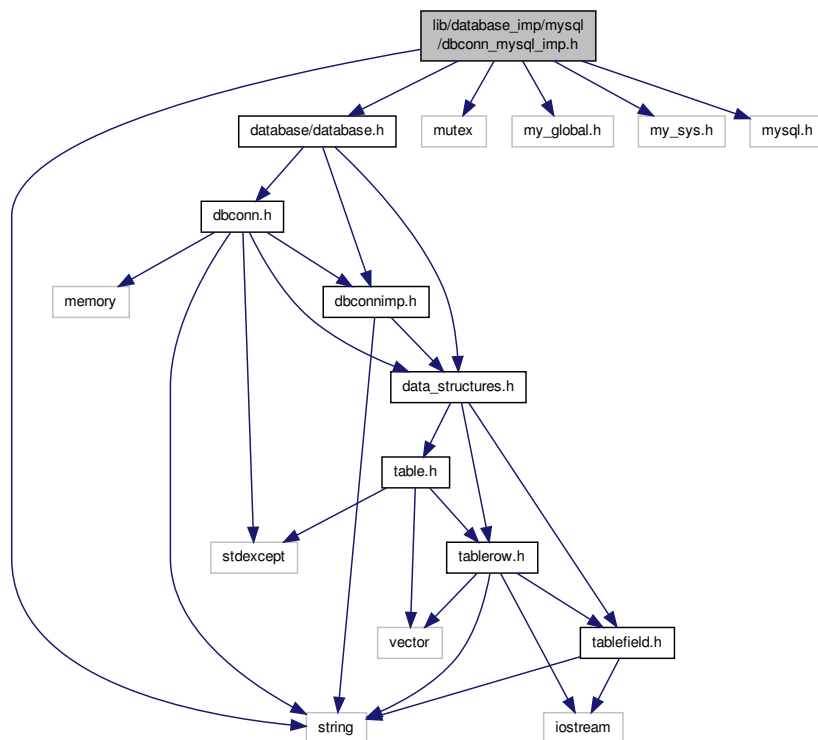
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.20 lib/database_imp/mysql/dbconn_mysql_imp.h File Reference

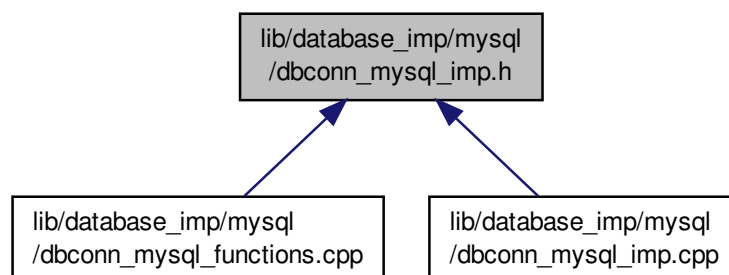
Interface to MySQL database connection implementation class.

```
#include <string>
#include <mutex>
#include "database/database.h"
#include <my_global.h>
#include <my_sys.h>
#include <mysql.h>
```

Include dependency graph for dbconn_mysql_imp.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [gldb::DBConnMySQL](#)

MySQL database implementation class.

10.20.1 Detailed Description

Interface to MySQL database connection implementation class.

Author

Paul Griffiths

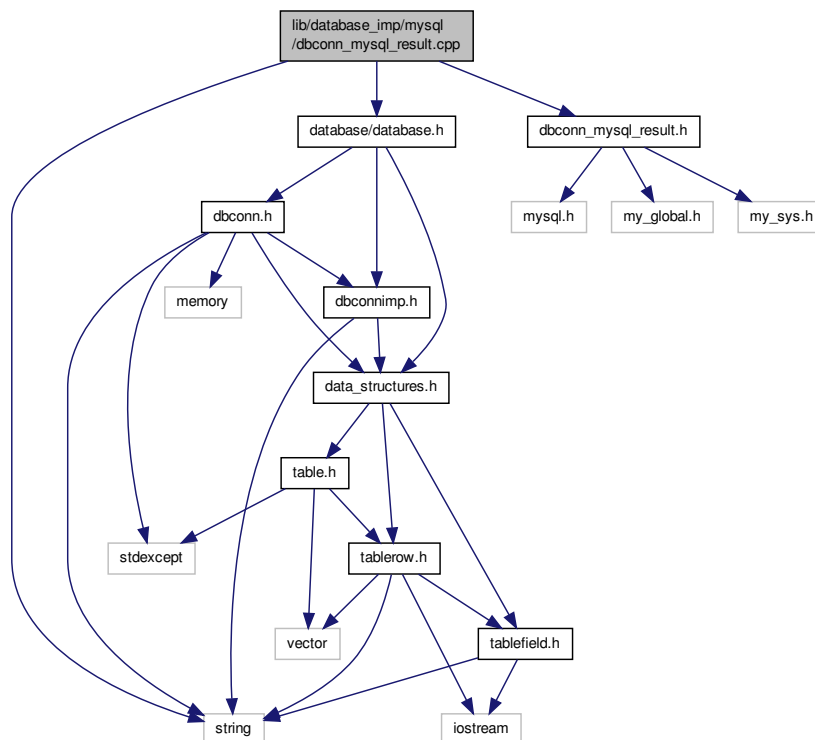
Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.21 lib/database_imp/mysql/dbconn_mysql_result.cpp File Reference

Implementation of MySQL result structure resource handle class.

```
#include <string>
#include "database/database.h"
#include "dbconn_mysql_result.h"
Include dependency graph for dbconn_mysql_result.cpp:
```



10.21.1 Detailed Description

Implementation of MySQL result structure resource handle class.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.22 lib/database_imp/mysql/dbconn_mysql_result.h File Reference

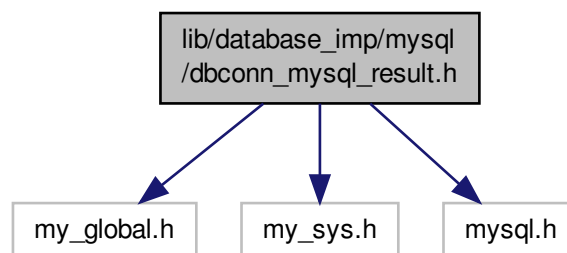
Interface to MySQL result structure resource handle class.

```
#include <my_global.h>
```

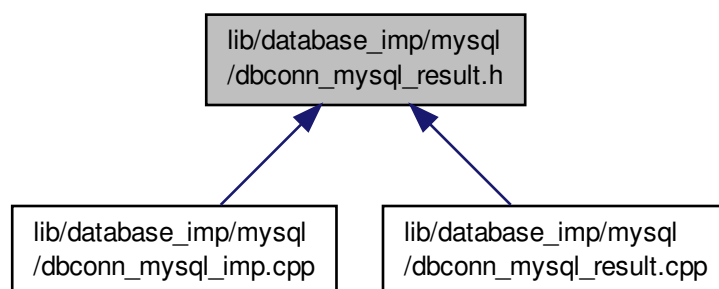
```
#include <my_sys.h>
```

```
#include <mysql.h>
```

Include dependency graph for dbconn_mysql_result.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `gldb::MySQLResult`
MySQL result structure class.

10.22.1 Detailed Description

Interface to MySQL result structure resource handle class.

Author

Paul Griffiths

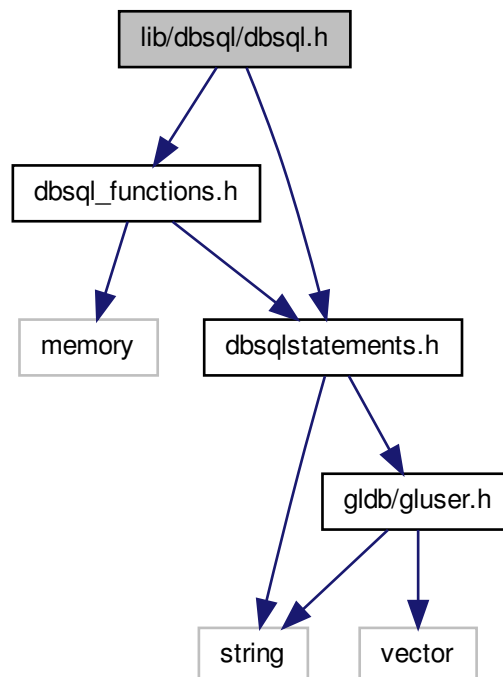
Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

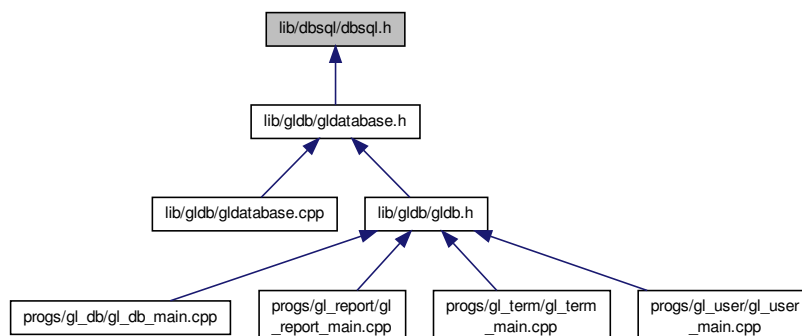
10.23 lib/dbsql/dbsql.h File Reference

User interface to DBSQL module.

```
#include "dbsql_functions.h"
#include "dbsqlstatements.h"
Include dependency graph for dbsql.h:
```



This graph shows which files directly or indirectly include this file:



10.23.1 Detailed Description

User interface to DBSQL module.

Author

Paul Griffiths

Copyright

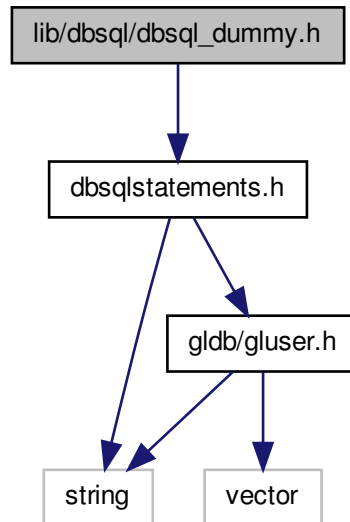
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.24 lib/dbsql/dbsql_dummy.h File Reference

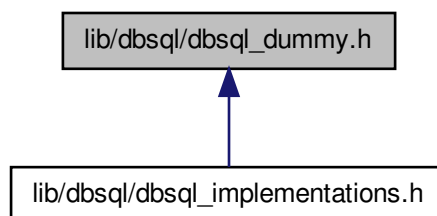
Interface to dummy SQL statement class.

```
#include "dbsqlstatements.h"
```

Include dependency graph for `dbsql_dummy.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [genleg::DBSQLDummy](#)
Dummy SQL statements class.

10.24.1 Detailed Description

Interface to dummy SQL statement class.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

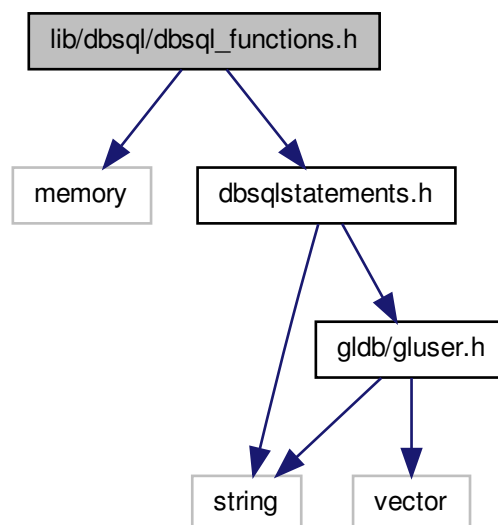
10.25 lib/dbsql/dbsql_functions.h File Reference

Interface to SQL module standalone functions.

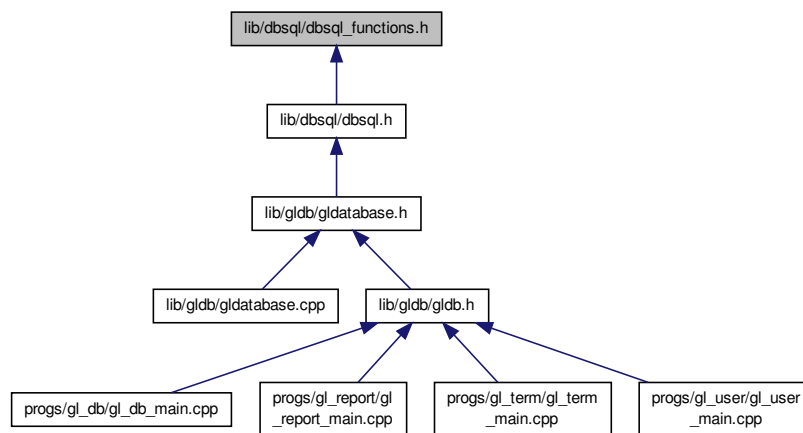
```
#include <memory>
```

```
#include "dbsqlstatements.h"
```

Include dependency graph for dbsql_functions.h:



This graph shows which files directly or indirectly include this file:



Functions

- `std::shared_ptr< const DBSQLStatements > genleg::get_sql_object ()`

Factory function for DBSQL objects.

10.25.1 Detailed Description

Interface to SQL module standalone functions.

Author

Paul Griffiths

Copyright

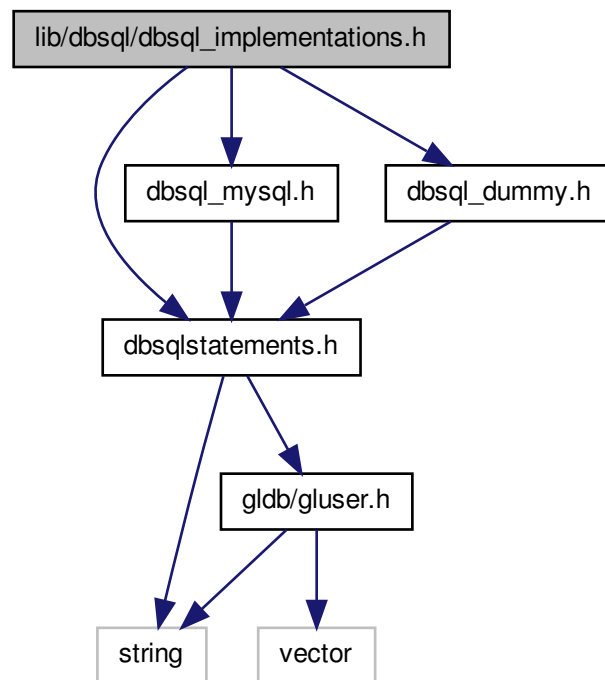
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.26 lib/dbsql/dbsql_implementations.h File Reference

Aggregation header for DBSQLStatements implementations.

```
#include "dbsqlstatements.h"
#include "dbsql_mysql.h"
#include "dbsql_dummy.h"
```


Include dependency graph for dbsql_implementations.h:



10.26.1 Detailed Description

Aggregation header for DBSqlStatements implementations.

Author

Paul Griffiths

Copyright

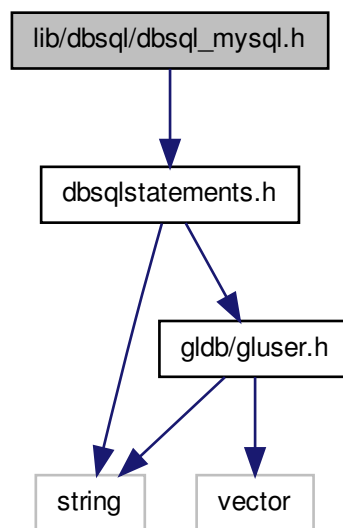
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.27 lib/dbsql/dbsql_mysql.h File Reference

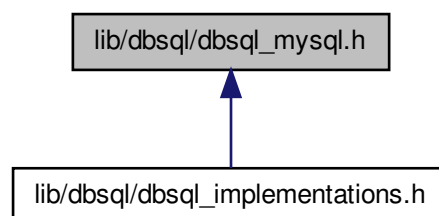
Interface to MySQL SQL statement class.

```
#include "dbsqlstatements.h"
```

Include dependency graph for `dbsql_mysql.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `genleg::DBSQLMySQL`

MySQL SQL statements class.

10.27.1 Detailed Description

Interface to MySQL SQL statement class.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

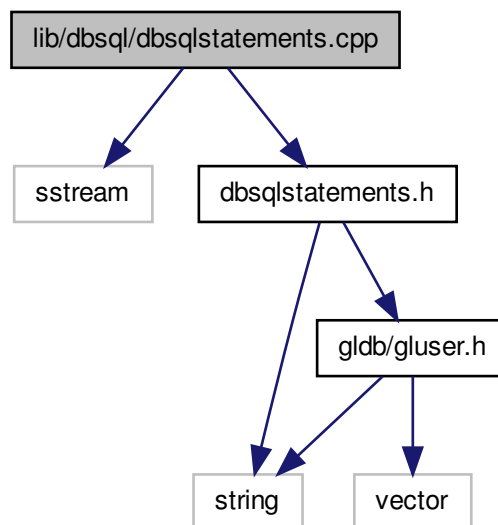
10.28 lib/dbsql/dbsqlstatements.cpp File Reference

Implementation of SQL statement class.

```
#include <sstream>
```

```
#include "dbsqlstatements.h"
```

Include dependency graph for dbsqlstatements.cpp:



10.28.1 Detailed Description

Implementation of SQL statement class.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

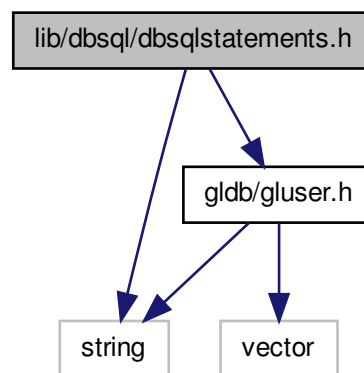
10.29 lib/dbsql/dbsqlstatements.h File Reference

Implementation of SQL module standalone functions.

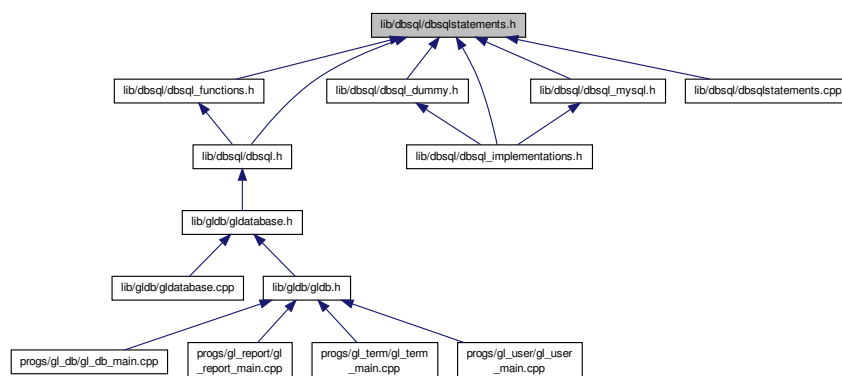
```
#include <string>
```

```
#include "gldb/gluser.h"
```

Include dependency graph for `dbsqlstatements.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [genleg::DBSQLStatements](#)
SQL statements class.

10.29.1 Detailed Description

Implementation of SQL module standalone functions. Interface to SQL statements class.

Author

Paul Griffiths

Copyright

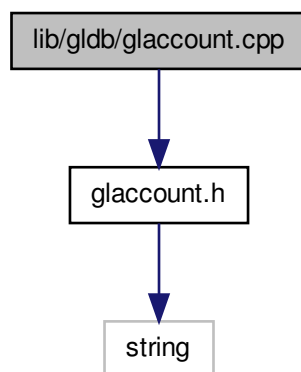
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.30 lib/gldb/glaccount.cpp File Reference

Implementation of nominal account class.

```
#include "glaccount.h"
```

Include dependency graph for glaccount.cpp:



10.30.1 Detailed Description

Implementation of nominal account class.

Author

Paul Griffiths

Copyright

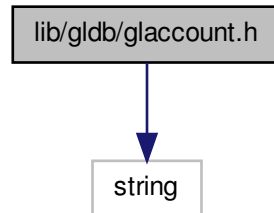
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.31 lib/gldb/glaccount.h File Reference

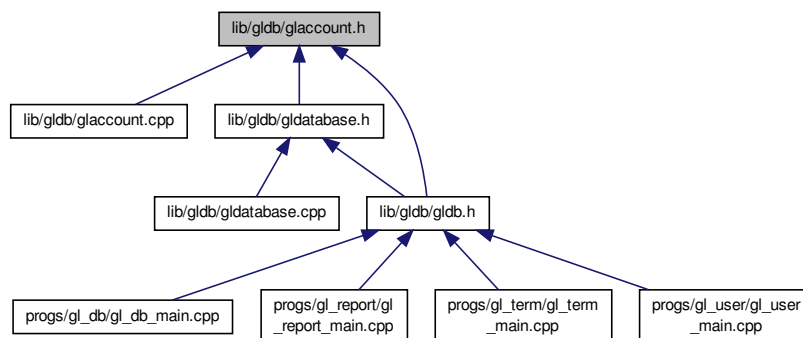
Interface to nominal account class.

```
#include <string>
```

Include dependency graph for glaccount.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [genleg::GLAccount](#)

Nominal account class.

10.31.1 Detailed Description

Interface to nominal account class.

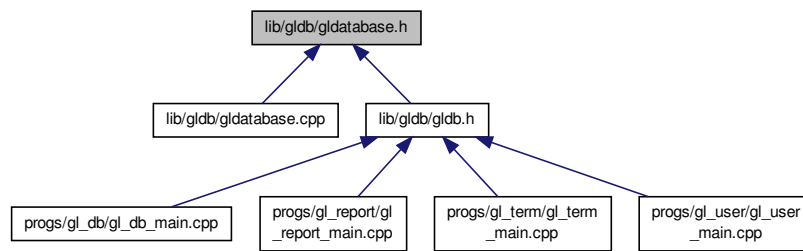
Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

This graph shows which files directly or indirectly include this file:



Classes

- class `genleg::GLDatabase`
General ledger database class.
- class `genleg::GLDBTransaction`
Database transaction RAII class.

10.33.1 Detailed Description

Interface to General Ledger database class.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.34 lib/gldb/gldb.h File Reference

User interface to General Ledger database module.

```

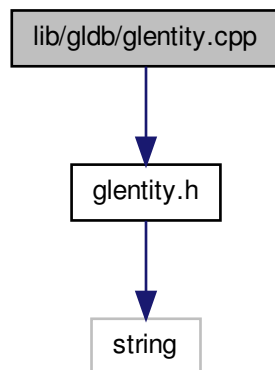
#include "glexception.h"
#include "gldatabase.h"
#include "gluser.h"
#include "glreport.h"
#include "gljournal.h"
#include "glentity.h"
#include "glaccount.h"
#include "glstanding.h"

```



```
#include "glentity.h"
```

Include dependency graph for glentity.cpp:



10.35.1 Detailed Description

Implementation of general ledger entity class.

Author

Paul Griffiths

Copyright

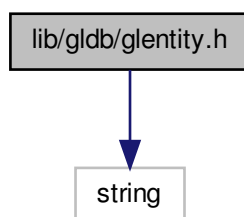
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.36 lib/gldb/glentity.h File Reference

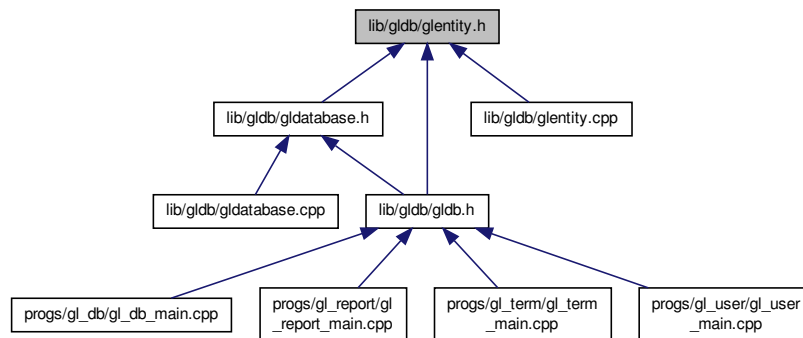
Interface to general ledger entity class.

```
#include <string>
```

Include dependency graph for glentity.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [genleg::GLEntity](#)

General ledger entity class.

10.36.1 Detailed Description

Interface to general ledger entity class.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

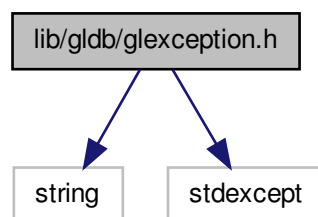
10.37 lib/gldb/glexception.h File Reference

Interface to General Ledger base exception class.

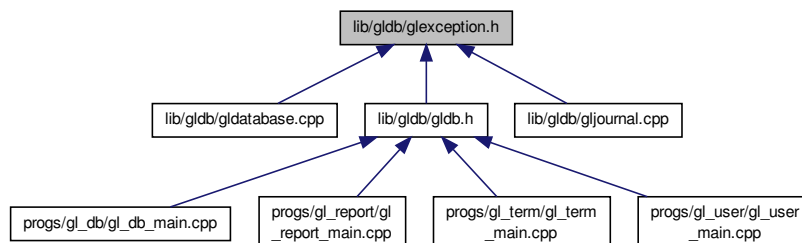
```
#include <string>
```

```
#include <stdexcept>
```

Include dependency graph for glexception.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [genleg::GLDBException](#)
Base general ledger database exception class.

10.37.1 Detailed Description

Interface to General Ledger base exception class.

Author

Paul Griffiths

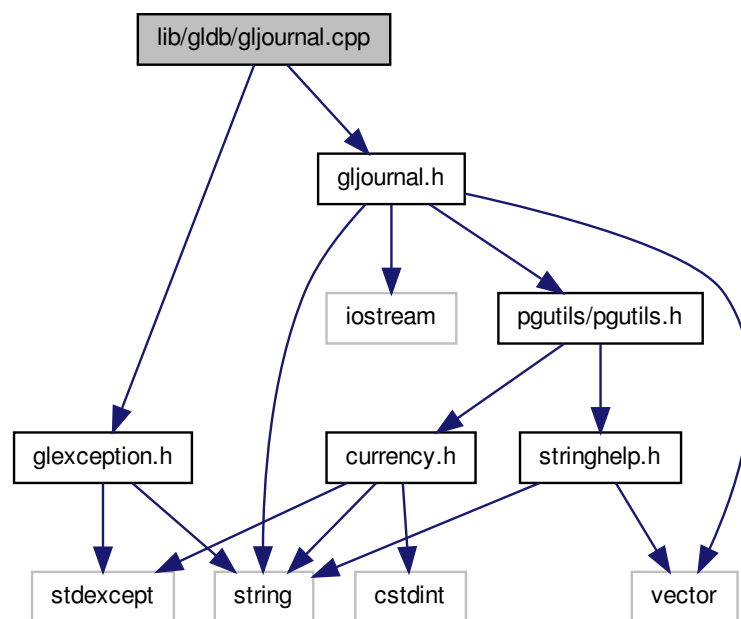
Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.38 lib/gldb/gljournal.cpp File Reference

Implementation of journal entry classes.

```
#include "gljournal.h"
#include "glexception.h"
Include dependency graph for gljournal.cpp:
```



10.38.1 Detailed Description

Implementation of journal entry classes.

Author

Paul Griffiths

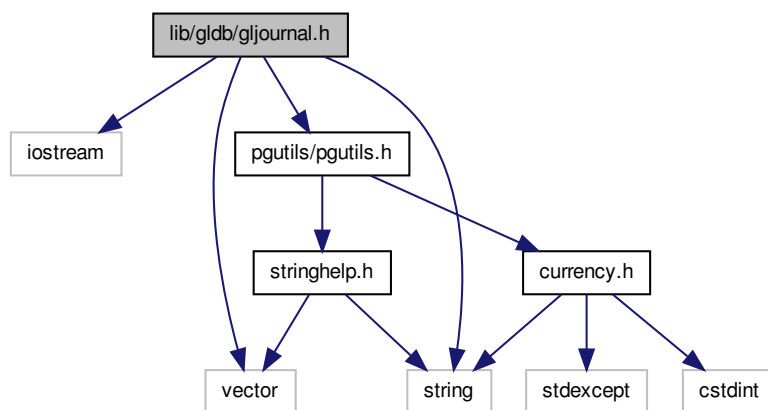
Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

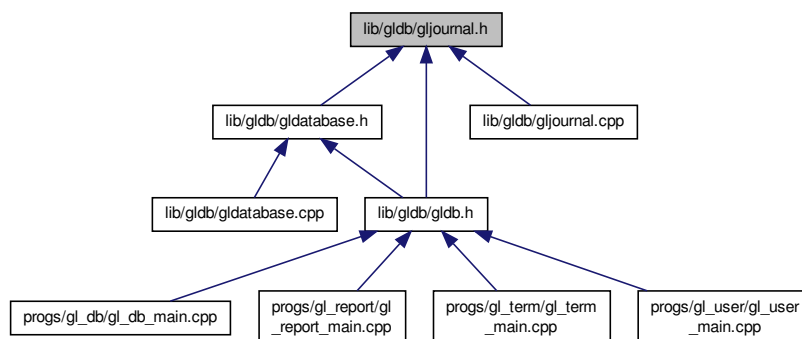
10.39 lib/gldb/gljournal.h File Reference

Interface to journal entry classes.

```
#include <iostream>
#include <vector>
#include <string>
#include "pgutils/pgutils.h"
Include dependency graph for gljournal.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `genleg::GLJELine`
Journal entry line class.
- class `genleg::GLJournal`
Journal entry class.

Functions

- `GLJournal genleg::journal_from_stream` (`std::istream &ifs`)
Returns a journal entry from a stream in a standard format.

Increments a vector of required column widths.

- static std::string `separator_row` (const std::vector< size_t > &widths)

Returns a decorated separator row for a table.

- static std::string `plain_row` (const TableRow &row, const std::vector< size_t > &widths)

Returns a row for a plain report.

- static std::string `decorated_row` (const TableRow &row, const std::vector< size_t > &widths)

Returns a row for a decorated report.

10.40.1 Detailed Description

Implementation of report class.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.41 lib/glib/glibreport.h File Reference

Interface to report class.

```
#include <string>
#include <database/database.h>
```


Functions

- `std::string genleg::plain_report_from_table` (const `gldb::Table` &table)
Creates a plain report from a table.
- `std::string genleg::decorated_report_from_table` (const `gldb::Table` &table)
Creates a decorated report from a table.
- `std::ostream & genleg::operator<<` (std::ostream &out, const `GLReport` &report)
Overridden << operator for printing a report.

10.41.1 Detailed Description

Interface to report class.

Author

Paul Griffiths

Copyright

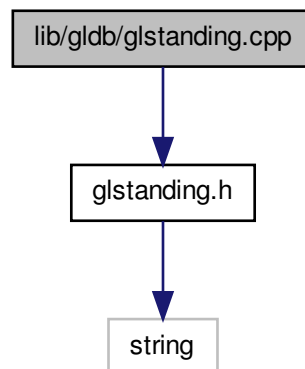
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.42 lib/gldb/glstanding.cpp File Reference

Implementation of general ledger standing data class.

```
#include "glstanding.h"
```

Include dependency graph for glstanding.cpp:



10.42.1 Detailed Description

Implementation of general ledger standing data class.

Author

Paul Griffiths

Copyright

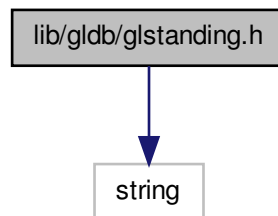
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.43 lib/gldb/glstanding.h File Reference

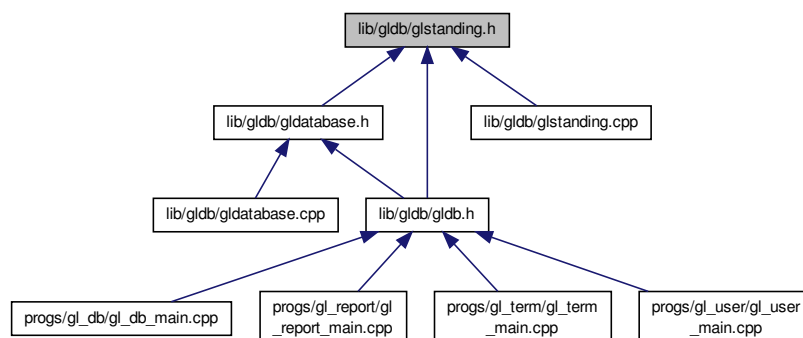
Interface to general ledger standing data class.

```
#include <string>
```

Include dependency graph for glstanding.h:



This graph shows which files directly or indirectly include this file:



Classes

- class `genleg::GLStandingData`

General ledger standing data class.

10.43.1 Detailed Description

Interface to general ledger standing data class.

Author

Paul Griffiths

Copyright

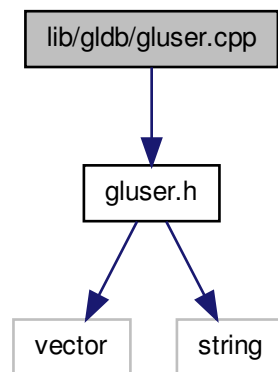
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.44 lib/gldb/gluser.cpp File Reference

Implementation of user class.

```
#include "gluser.h"
```

Include dependency graph for gluser.cpp:



10.44.1 Detailed Description

Implementation of user class.

Author

Paul Griffiths

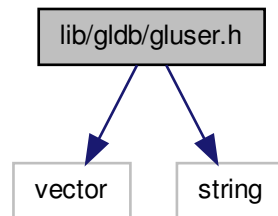
Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

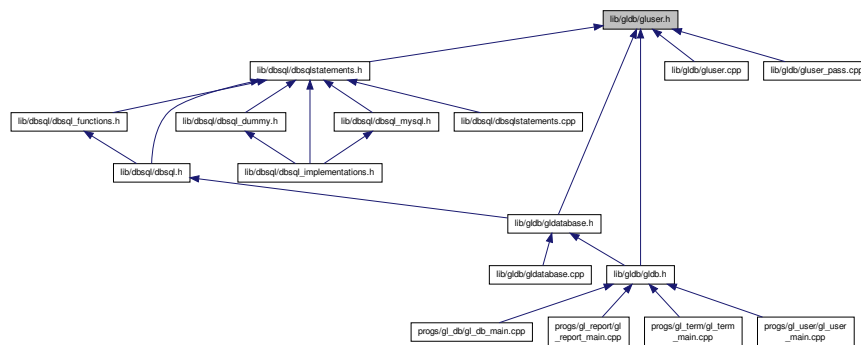
10.45 lib/gldb/gluser.h File Reference

Interface to user class.

```
#include <vector>
#include <string>
Include dependency graph for gluser.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `genleg::GLUser`
General ledger user class.

10.45.1 Detailed Description

Interface to user class.

Author

Paul Griffiths

Copyright

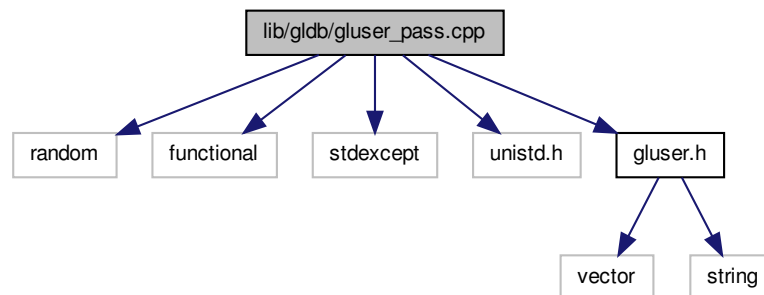
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.46 lib/gldb/gluser_pass.cpp File Reference

Implementation of password functions for user class.

```
#include <random>
#include <functional>
#include <stdexcept>
#include <unistd.h>
#include "gluser.h"
```

Include dependency graph for gluser_pass.cpp:



Macros

- `#define _XOPEN_SOURCE 600`

Functions

- static `std::string generate_salt ()`
Generates a random two-character salt for crypt()

10.46.1 Detailed Description

Implementation of password functions for user class.

Todo Implement a better form of password encryption. In particular, these functions are not re-entrant, and only use the first 8 characters of the password.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.46.2 Macro Definition Documentation

10.46.2.1 `#define _XOPEN_SOURCE 600`

UNIX feature test macro

10.46.3 Function Documentation

10.46.3.1 `static std::string generate_salt () [static]`

Generates a random two-character salt for crypt()

Returns

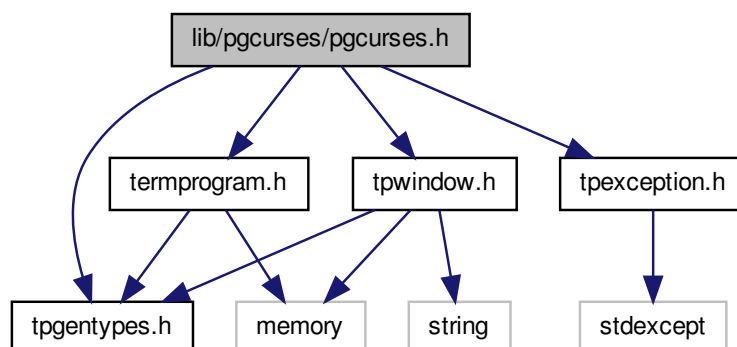
The two-character salt.

10.47 lib/pgcurses/pgcurses.h File Reference

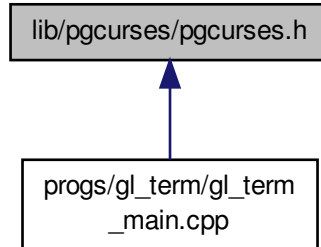
User interface to curses wrapper library.

```
#include "tpgentypes.h"
#include "tpexception.h"
#include "termprogram.h"
#include "tpwindow.h"
```

Include dependency graph for pgcurses.h:



This graph shows which files directly or indirectly include this file:



10.47.1 Detailed Description

User interface to curses wrapper library.

Author

Paul Griffiths

Copyright

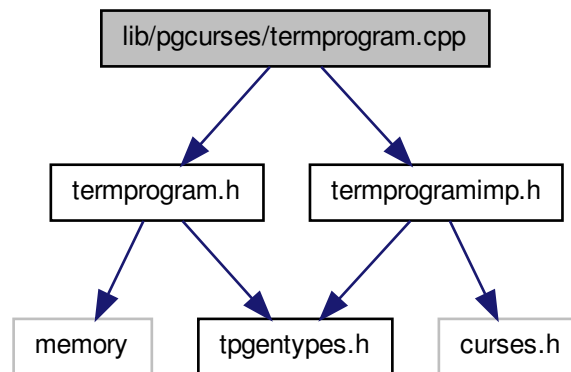
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.48 lib/pgcurses/termprogram.cpp File Reference

Implementation of main terminal program class.

```
#include "termprogram.h"
#include "termprogramimp.h"
```

Include dependency graph for `termprogram.cpp`:



10.48.1 Detailed Description

Implementation of main terminal program class.

Author

Paul Griffiths

Copyright

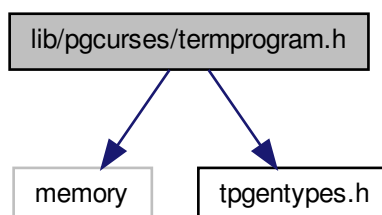
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.49 lib/pgcurses/termprogram.h File Reference

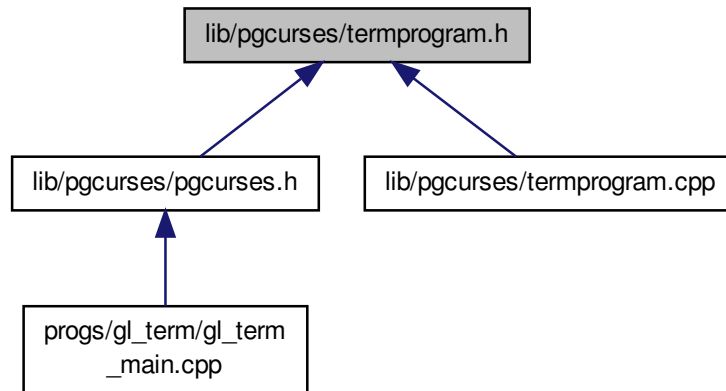
Interface to main terminal program class.

```
#include <memory>
#include "tpgentypes.h"
```

Include dependency graph for `termprogram.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `pgcurses::TermProgram`

Main terminal program class.

10.49.1 Detailed Description

Interface to main terminal program class.

Author

Paul Griffiths

Copyright

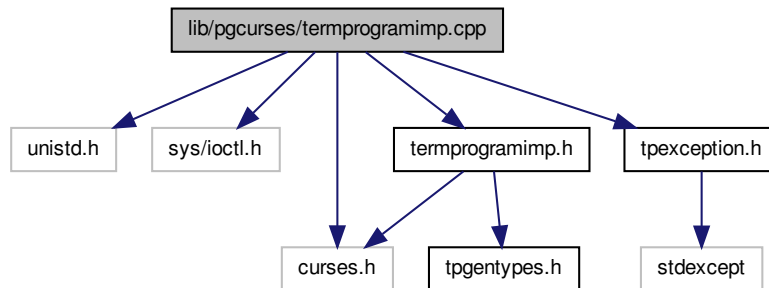
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.50 lib/pgcurses/termprogramimp.cpp File Reference

Implementation of main terminal program implementation class.

```
#include <unistd.h>
#include <sys/ioctl.h>
#include <curses.h>
#include "termprogramimp.h"
#include "tpexception.h"
```

Include dependency graph for `termprogramimp.cpp`:



10.50.1 Detailed Description

Implementation of main terminal program implementation class.

Author

Paul Griffiths

Copyright

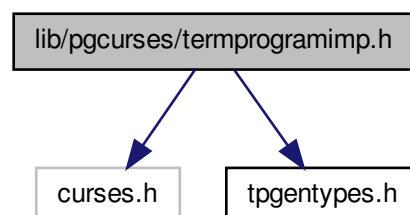
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.51 lib/pgcurses/termprogramimp.h File Reference

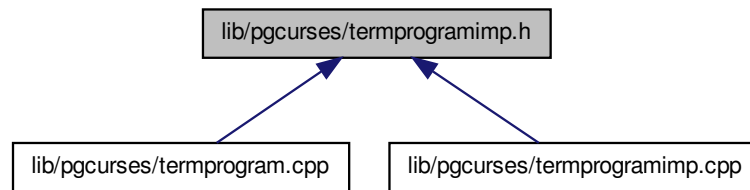
Interface to main terminal program implementation class.

```
#include <curses.h>
#include "tpgentypes.h"
```

Include dependency graph for `termprogramimp.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `pgcurses::TermProgramImp`
Main terminal program implementation class.

10.51.1 Detailed Description

Interface to main terminal program implementation class.

Author

Paul Griffiths

Copyright

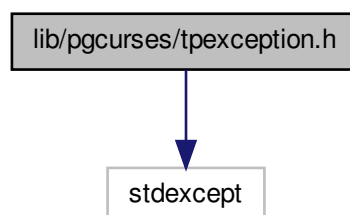
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.52 lib/pgcurses/tpexception.h File Reference

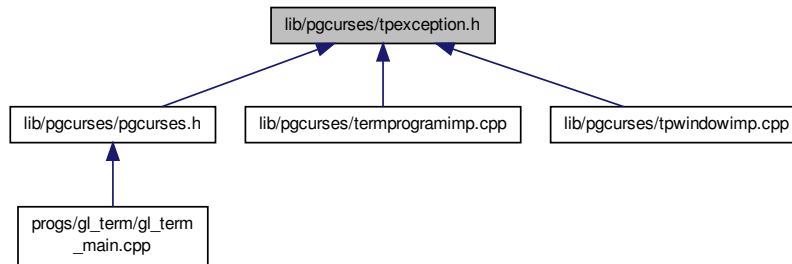
Interface to terminal program exception class.

```
#include <stdexcept>
```

Include dependency graph for `tpexception.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class [pgcurses::TPEException](#)
Base terminal program exception class.

10.52.1 Detailed Description

Interface to terminal program exception class.

Author

Paul Griffiths

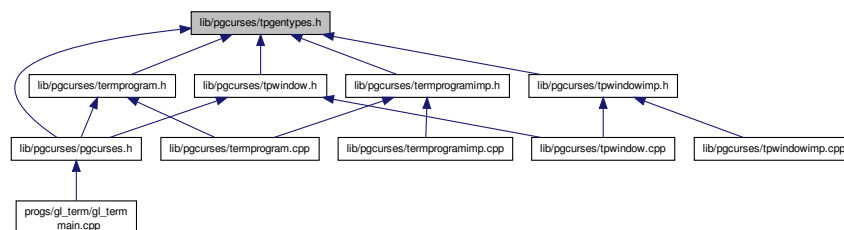
Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.53 lib/pgcurses/tpgentypes.h File Reference

Interface to terminal program general types.

This graph shows which files directly or indirectly include this file:



Classes

- struct [pgcurses::Point](#)

- Point* structure.
- struct `pgcurses::Size`
Size structure.
- struct `pgcurses::Rectangle`
Rectangle structure.

10.53.1 Detailed Description

Interface to terminal program general types.

Author

Paul Griffiths

Copyright

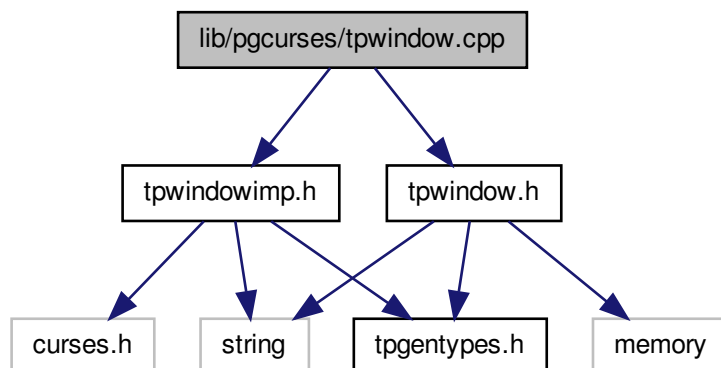
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.54 lib/pgcurses/tpwindow.cpp File Reference

Implementation of generic curses window class.

```
#include "tpwindow.h"  
#include "tpwindowimp.h"
```

Include dependency graph for tpwindow.cpp:



10.54.1 Detailed Description

Implementation of generic curses window class.

Author

Paul Griffiths

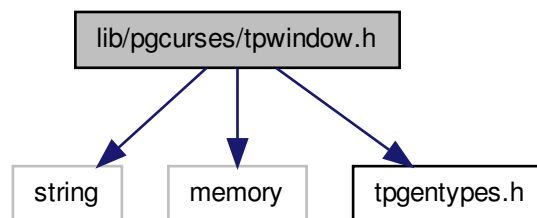
Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

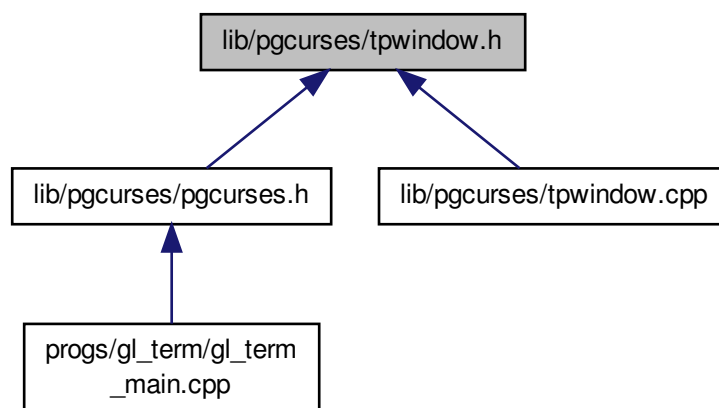
10.55 lib/pgcurses/tpwindow.h File Reference

Interface to generic curses window class.

```
#include <string>
#include <memory>
#include "tpgentypes.h"
Include dependency graph for tpwindow.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- class `pgcurses::TPWindow`
Generic window class.

10.55.1 Detailed Description

Interface to generic curses window class.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

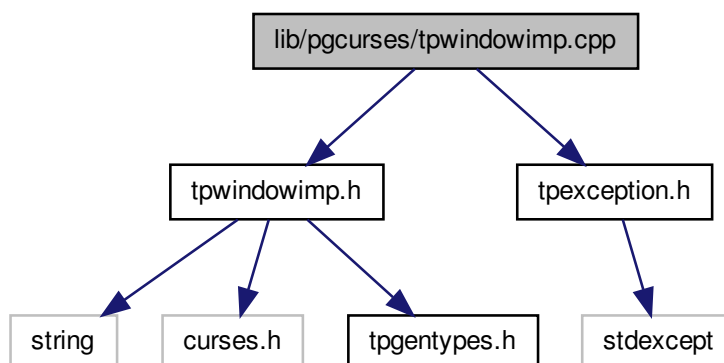
10.56 lib/pgcurses/tpwindowimp.cpp File Reference

Implementation of generic curses window impl class.

```
#include "tpwindowimp.h"
```

```
#include "tpexception.h"
```

Include dependency graph for tpwindowimp.cpp:



10.56.1 Detailed Description

Implementation of generic curses window impl class.

Author

Paul Griffiths

Copyright

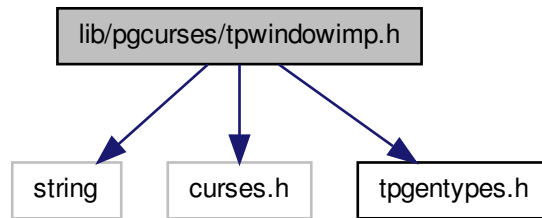
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.57 lib/pgcurses/tpwindowimp.h File Reference

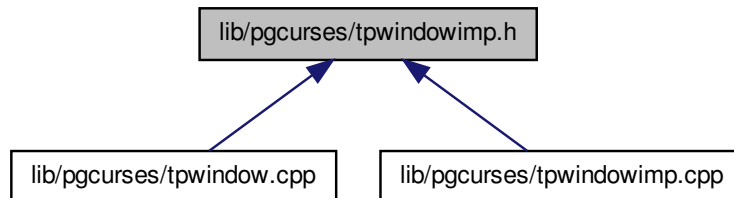
Interface to generic curses window impl class.

```
#include <string>
#include <curses.h>
#include "tpgentypes.h"
```

Include dependency graph for tpwindowimp.h:



This graph shows which files directly or indirectly include this file:



Classes

- class [pgcurses::TPWindowImp](#)
Terminal program window implementation class.

10.57.1 Detailed Description

Interface to generic curses window impl class.

Author

Paul Griffiths

Copyright

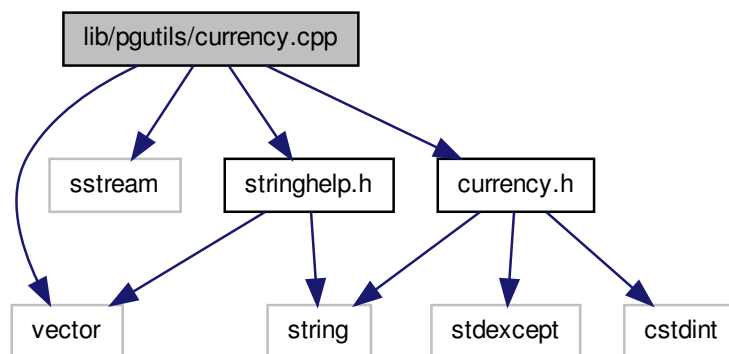
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.58 lib/pgutils/currency.cpp File Reference

Implementation of currency amount class.

```
#include <vector>
#include <sstream>
#include "currency.h"
#include "stringhelp.h"
```

Include dependency graph for currency.cpp:



10.58.1 Detailed Description

Implementation of currency amount class.

Author

Paul Griffiths

Copyright

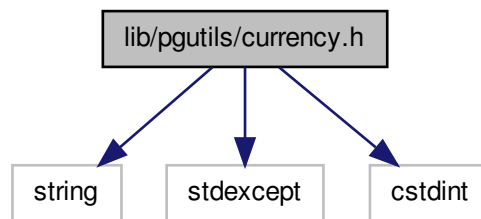
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.59 lib/pgutils/currency.h File Reference

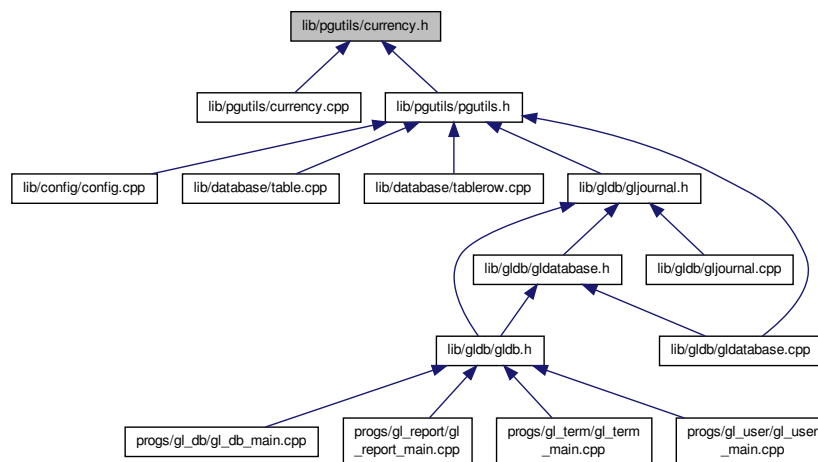
Interface to currency amount class.

```
#include <string>
#include <stdexcept>
#include <cstdint>
```

Include dependency graph for `currency.h`:



This graph shows which files directly or indirectly include this file:



Classes

- class `pgutils::CurrencyException`
Base `Currency` exception class.
- class `pgutils::Currency`
`Currency` amount class.

Functions

- `Currency pgutils::operator+ (Currency lhs, const Currency &rhs)`
`Currency` addition operator.
- `Currency pgutils::operator- (Currency lhs, const Currency &rhs)`
`Currency` subtraction operator.
- `bool pgutils::operator== (const Currency &lhs, const Currency &rhs)`
`Currency` equality comparison operator.
- `bool pgutils::operator!= (const Currency &lhs, const Currency &rhs)`

- *Currency inequality comparison operator.*
 • bool `pgutils::operator<` (const `Currency` &lhs, const `Currency` &rhs)
Currency less than comparison operator.
- bool `pgutils::operator>` (const `Currency` &lhs, const `Currency` &rhs)
Currency greater than comparison operator.
- bool `pgutils::operator<=` (const `Currency` &lhs, const `Currency` &rhs)
Currency less than or equal to comparison operator.
- bool `pgutils::operator>=` (const `Currency` &lhs, const `Currency` &rhs)
Currency greater than or equal to comparison operator.
- `Currency pgutils::currency_from_string` (const std::string &s)
Creates a currency amount from a string representation.

10.59.1 Detailed Description

Interface to currency amount class.

Author

Paul Griffiths

Copyright

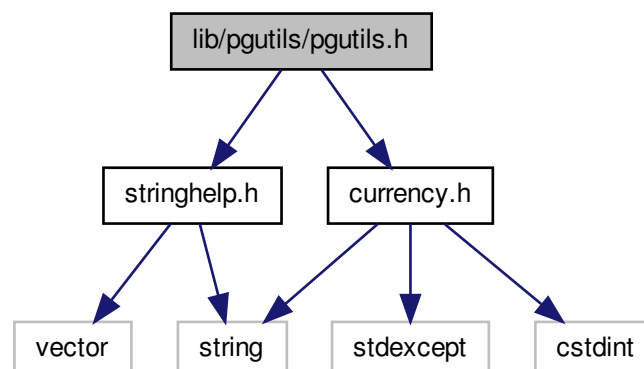
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.60 lib/pgutils/pgutils.h File Reference

Aggregate interface to general utility functions.

```
#include "stringhelp.h"
#include "currency.h"
```

Include dependency graph for pgutils.h:



10.61.1 Detailed Description

Implementation of string helper functions.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

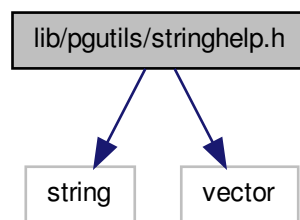
10.62 lib/pgutils/stringhelp.h File Reference

Interface to string helper functions.

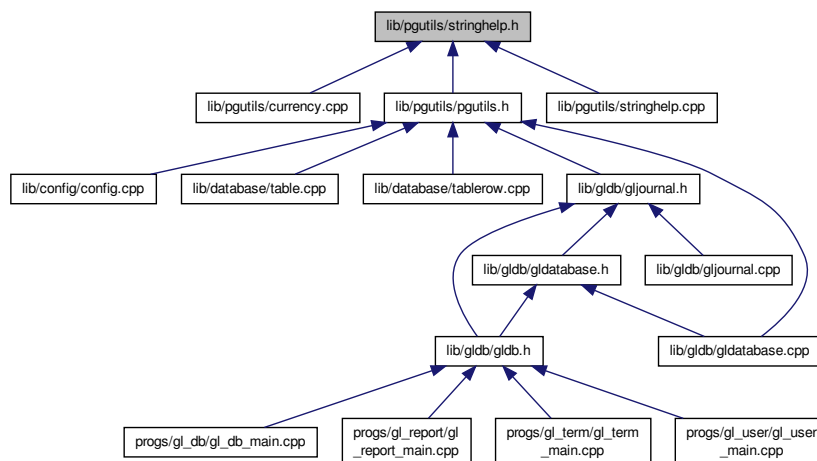
```
#include <string>
```

```
#include <vector>
```

Include dependency graph for stringhelp.h:



This graph shows which files directly or indirectly include this file:



Functions

- `std::string & pgutils::trim_front` (`std::string &s`)
Trims leading whitespace from a string.
- `std::string & pgutils::trim_back` (`std::string &s`)
Trims trailing whitespace from a string.
- `std::string & pgutils::trim` (`std::string &s`)
Trims leading and trailing whitespace from a string.
- `std::vector< std::string > pgutils::split` (`const std::string &s, const char delim`)
Splits a delimited string into tokens.
- `std::vector< std::string > & pgutils::split` (`std::vector< std::string > &vec, const std::string &s, const char delim`)
Splits a delimited string into tokens.
- `bool pgutils::next_content_line` (`std::istream &if, std::string &s`)
Gets the next content line from a stream.
- `std::vector< std::string > & pgutils::content_lines` (`std::vector< std::string > &vec, std::istream &if`)
Populates a vector of content lines from a stream.
- `std::vector< std::vector< std::string > > & pgutils::split_lines` (`std::vector< std::vector< std::string > > &vec, std::istream &if, const char delim`)
Populates a vector of vectors of fields from a stream.
- `std::string & pgutils::join` (`const std::vector< std::string > &vec, std::string &s, const char delim`)
Joins a vector of strings into a delimited line.
- `bool pgutils::replace` (`std::string &str, const std::string &from, const std::string &to`)
Replaces a substring with another string.

10.62.1 Detailed Description

Interface to string helper functions.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.63 progs/gl_db/gl_db_main.cpp File Reference

Main functionality for gl_db program.

```
#include <iostream>
#include "gldb/gldb.h"
#include "config/config.h"
```


Variables

- `static const char * progname = "gl_report"`
Static variable for program name.

10.64.1 Detailed Description

Main functionality for gl_report program.

Author

Paul Griffiths

Copyright

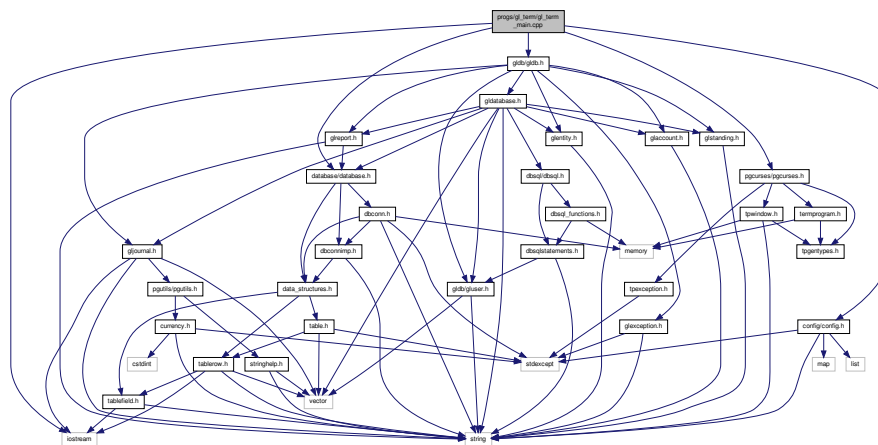
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.65 progs/gl_term/gl_term_main.cpp File Reference

Main functionality for gl_term program.

```
#include <iostream>
#include "gldb/gldb.h"
#include "database/database.h"
#include "config/config.h"
#include "pgcurses/pgcurses.h"
```

Include dependency graph for gl_term_main.cpp:



Functions

- static void `set_configuration` (`Config` &config, int argc, char *argv[])
Sets program configuration options.
- static bool `check_help_and_version` (const `Config` &config)
Prints help or version messages if requested.
- static bool `check_db_parameters` (const `Config` &config)
Checks if database, hostname and username were provided.

- static void `print_usage_message` ()
Prints a program usage message.
- static void `print_version_message` ()
Prints a program version message.
- static void `print_help_message` ()
Prints a program help message.
- static std::string `login` (void)
Gets a password from the terminal.
- int `main` (int argc, char *argv[])
Main function.

Variables

- static const char * `progrname` = "gl_term"
Static variable for program name.

10.65.1 Detailed Description

Main functionality for gl_term program. This is a curses based terminal interface to the general ledger.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

10.66 progs/gl_user/gl_user_main.cpp File Reference

Main functionality for gl_user program.

```
#include <iostream>
#include "gldb/gldb.h"
#include "database/database.h"
#include "config/config.h"
```


10.66.1 Detailed Description

Main functionality for gl_user program.

Author

Paul Griffiths

Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

Index

- ~Config
 - genleg::Config, [43](#)
- ~DBConnDummy
 - gldb::DBConnDummy, [61](#)
- ~DBConnImp
 - gldb::DBConnImp, [63](#)
- ~DBConnMySQL
 - gldb::DBConnMySQL, [66](#)
- ~DBSQLStatements
 - genleg::DBSQLStatements, [71](#)
- ~GLDBTransaction
 - genleg::GLDBTransaction, [87](#)
- ~GLDatabase
 - genleg::GLDatabase, [80](#)
- ~GLUser
 - genleg::GLUser, [97](#)
- ~MySQLResult
 - gldb::MySQLResult, [100](#)
- ~TPWindow
 - pgcurses::TPWindow, [134](#)
- ~TPWindowImp
 - pgcurses::TPWindowImp, [137](#)
- ~Table
 - gldb::Table, [106](#)
- ~TableField
 - gldb::TableField, [116](#)
- ~TableRow
 - gldb::TableRow, [125](#)
- ~TermProgram
 - pgcurses::TermProgram, [129](#)
- ~TermProgramImp
 - pgcurses::TermProgramImp, [131](#)
- _XOPEN_SOURCE
 - config_getopt.cpp, [142](#)
 - gluser_pass.cpp, [194](#)
- account
 - genleg::GLJELine, [90](#)
- account_by_name
 - genleg::DBSQLStatements, [71](#)
- add_cmdline_option
 - genleg::Config, [44](#)
- amount
 - genleg::GLJELine, [90](#)
- append_field
 - gldb::TableRow, [125](#), [126](#)
- append_record
 - gldb::Table, [107](#)
- backend
 - genleg::GLDatabase, [80](#)
- begin
 - genleg::GLJournal, [91](#), [92](#)
 - gldb::Table, [107](#)
 - gldb::TableRow, [126](#)
- boolstring_to_bool
 - gldatabase.cpp, [177](#)
- check_db_parameters
 - Database program., [33](#)
 - Reporting program., [35](#)
 - Terminal access program., [37](#)
 - User administration program., [39](#)
- check_help_and_version
 - Database program., [33](#)
 - Reporting program., [35](#)
 - Terminal access program., [37](#)
 - User administration program., [40](#)
- check_password
 - genleg::GLUser, [97](#)
- check_user_password
 - User administration program., [40](#)
- Config
 - genleg::Config, [43](#)
- config_getopt.cpp
 - _XOPEN_SOURCE, [142](#)
- ConfigBadConfigFile
 - genleg::ConfigBadConfigFile, [46](#)
- ConfigBadOption
 - genleg::ConfigBadOption, [47](#)
- ConfigCouldNotOpenFile
 - genleg::ConfigCouldNotOpenFile, [49](#)
- ConfigException
 - genleg::ConfigException, [50](#)
- ConfigOptionNotSet
 - genleg::ConfigOptionNotSet, [51](#)
- content_lines
 - General purpose utilities., [27](#)
- create_entity
 - genleg::GLDatabase, [80](#)
- create_from_file
 - gldb::Table, [107](#)
- create_structure
 - genleg::GLDatabase, [80](#)
- create_table
 - genleg::DBSQLStatements, [71](#)
- create_user
 - genleg::GLDatabase, [80](#)
- create_view
 - genleg::DBSQLStatements, [72](#)

- Currency
 - pgutils::Currency, 52
- currency_from_string
 - General purpose utilities., 27
- CurrencyException
 - pgutils::CurrencyException, 54
- current_trial_balance_report
 - genleg::GLDatabase, 81
- currentttb
 - genleg::DBSQLStatements, 72
- currentttb_by_entity
 - genleg::DBSQLStatements, 72
- DBConn
 - gldb::DBConn, 56
- DBConnCouldNotConnect
 - gldb::DBConnCouldNotConnect, 58
- DBConnCouldNotQuery
 - gldb::DBConnCouldNotQuery, 59
- DBConnDummy
 - gldb::DBConnDummy, 60, 61
- DBConnException
 - gldb::DBConnException, 62
- DBConnImp
 - gldb::DBConnImp, 63
- DBConnMySQL
 - gldb::DBConnMySQL, 66
- DBSQLStatements
 - genleg::DBSQLStatements, 71
- Database interaction module, 18
 - get_connection, 19
 - get_database_type, 19
 - get_field_names, 19
 - get_row, 19
- Database program., 33
 - check_db_parameters, 33
 - check_help_and_version, 33
 - login, 34
 - main, 34
 - set_configuration, 34
- decorated_report_from_table
 - General Ledger database module., 23
- decorated_row
 - General Ledger database module., 23
- destroy_structure
 - genleg::GLDatabase, 81
- drop_table
 - genleg::DBSQLStatements, 72
- drop_view
 - genleg::DBSQLStatements, 72
- enable_user
 - User administration program., 40
- enabled
 - genleg::GLUser, 97
- end
 - genleg::GLJournal, 92
 - gldb::Table, 107, 108
 - gldb::TableRow, 126
- entity_by_id
 - genleg::DBSQLStatements, 73
- entity_by_name
 - genleg::DBSQLStatements, 73
- expand
 - pgutils::Currency, 52
- firstname
 - genleg::GLUser, 97
- GLAccount
 - genleg::GLAccount, 77
- GLDBException
 - genleg::GLDBException, 85
- GLDBTransaction
 - genleg::GLDBTransaction, 86
- GLDatabase
 - genleg::GLDatabase, 79
- GLEntity
 - genleg::GLEntity, 88
- GLJELine
 - genleg::GLJELine, 89
- GLJournal
 - genleg::GLJournal, 91
- GLReport
 - genleg::GLReport, 93
- GLStandingData
 - genleg::GLStandingData, 94
- GLUser
 - genleg::GLUser, 96
- General Ledger database module., 22
 - decorated_report_from_table, 23
 - decorated_row, 23
 - grow_widths, 23
 - max_column_widths, 23
 - plain_report_from_table, 23
 - plain_row, 24
 - separator_row, 24
- General purpose utilities., 26
 - content_lines, 27
 - currency_from_string, 27
 - join, 27
 - next_content_line, 27
 - operator<, 28
 - operator<=, 29
 - operator>, 29
 - operator>=, 29
 - operator+, 28
 - operator-, 28
 - operator==, 29
 - replace, 30
 - split, 30
 - split_lines, 31
 - trim, 31
 - trim_back, 31
 - trim_front, 31
- generate_salt
 - gluser_pass.cpp, 194
- genleg::Config, 43

- ~Config, 43
- add_cmdline_option, 44
- Config, 43
- is_set, 44
- m_opts_set, 45
- m_opts_supp, 45
- populate_from_cmdline, 44
- populate_from_file, 45
- genleg::ConfigBadConfigFile, 45
 - ConfigBadConfigFile, 46
- genleg::ConfigBadOption, 47
 - ConfigBadOption, 47
- genleg::ConfigCouldNotOpenFile, 48
 - ConfigCouldNotOpenFile, 49
- genleg::ConfigException, 49
 - ConfigException, 50
- genleg::ConfigOptionNotSet, 50
 - ConfigOptionNotSet, 51
- genleg::DBSQLDummy, 68
- genleg::DBSQLMySQL, 68
- genleg::DBSQLStatements, 69
 - ~DBSQLStatements, 71
 - account_by_name, 71
 - create_table, 71
 - create_view, 72
 - currenttb, 72
 - currenttb_by_entity, 72
 - DBSQLStatements, 71
 - drop_table, 72
 - drop_view, 72
 - entity_by_id, 73
 - entity_by_name, 73
 - get_perms, 73
 - grant, 73
 - je_by_id, 74
 - jelines_by_id, 74
 - listusers, 74
 - post_je, 74
 - post_je_line, 75
 - revoke, 75
 - standing_data, 75
 - update_user, 75
 - user_by_id, 75
 - user_by_username, 76
- genleg::GLAccount, 76
 - GLAccount, 77
 - m_description, 77
 - m_enabled, 77
- genleg::GLDBException, 85
 - GLDBException, 85
- genleg::GLDBTransaction, 86
 - ~GLDBTransaction, 87
 - GLDBTransaction, 86
 - m_commit, 87
 - m_dbc, 87
- genleg::GLDatabase, 77
 - ~GLDatabase, 80
 - backend, 80
 - create_entity, 80
 - create_structure, 80
 - create_user, 80
 - current_trial_balance_report, 81
 - destroy_structure, 81
 - GLDatabase, 79
 - get_account_by_name, 81
 - get_entity_by_id, 81
 - get_entity_by_name, 82
 - get_je_by_id, 82
 - get_standing_data, 82
 - get_user_by_id, 82
 - get_user_by_username, 83
 - grant, 83
 - je_report, 83
 - list_users_report, 83
 - load_sample_data, 83
 - m_dbc, 84
 - m_sql, 85
 - m_tables, 85
 - m_views, 85
 - post_journal, 84
 - report, 84
 - revoke, 84
 - standing_data_report, 84
 - update_user, 84
- genleg::GLEntity, 87
 - GLEntity, 88
 - m_aggregate, 88
 - m_enabled, 88
 - m_name, 88
 - m_parent, 88
 - m_shortcode, 88
- genleg::GLJELine, 89
 - account, 90
 - amount, 90
 - GLJELine, 89
 - m_acct, 90
 - m_amount, 90
- genleg::GLJournal, 90
 - begin, 91, 92
 - end, 92
 - GLJournal, 91
 - m_entity, 92
 - m_id, 92
 - m_lines, 92
 - m_memo, 92
 - m_period, 92
 - m_source, 92
 - m_user, 93
 - m_year, 93
- genleg::GLReport, 93
 - GLReport, 93
 - m_headers, 93
 - m_report_text, 93
 - m_title, 94
- genleg::GLStandingData, 94
 - GLStandingData, 94

- m_num_periods, 94
 - m_period, 94
 - m_year, 95
- genleg::GLUser, 95
 - ~GLUser, 97
 - check_password, 97
 - enabled, 97
 - firstname, 97
 - GLUser, 96
 - id, 97
 - lastname, 97
 - m_enabled, 99
 - m_firstname, 99
 - m_id, 99
 - m_lastname, 99
 - m_pass_hash, 99
 - m_pass_salt, 99
 - m_perms, 99
 - m_username, 99
 - pass_hash, 97
 - pass_salt, 98
 - permissions, 98
 - set_enabled, 98
 - set_firstname, 98
 - set_lastname, 98
 - set_password, 98
 - set_username, 99
 - username, 99
- get_account_by_name
 - genleg::GLDatabase, 81
- get_connection
 - Database interaction module, 19
- get_database_type
 - Database interaction module, 19
- get_entity_by_id
 - genleg::GLDatabase, 81
- get_entity_by_name
 - genleg::GLDatabase, 82
- get_field
 - gldb::Table, 108
- get_field_names
 - Database interaction module, 19
- get_headers
 - gldb::Table, 108
- get_je_by_id
 - genleg::GLDatabase, 82
- get_perms
 - genleg::DBSQLStatements, 73
- get_row
 - Database interaction module, 19
- get_standing_data
 - genleg::GLDatabase, 82
- get_user
 - User administration program., 40
- get_user_by_id
 - genleg::GLDatabase, 82
- get_user_by_username
 - genleg::GLDatabase, 83
- gldatabase.cpp
 - boolstring_to_bool, 177
- gldb::DBConn, 54
 - DBConn, 56
 - last_auto_increment, 56
 - m_imp, 57
 - operator=, 56
 - query, 56
 - select, 56
- gldb::DBConnCouldNotConnect, 57
 - DBConnCouldNotConnect, 58
- gldb::DBConnCouldNotQuery, 58
 - DBConnCouldNotQuery, 59
- gldb::DBConnDummy, 59
 - ~DBConnDummy, 61
 - DBConnDummy, 60, 61
 - operator=, 61
 - query, 61
 - select, 61
- gldb::DBConnException, 62
 - DBConnException, 62
- gldb::DBConnImp, 62
 - ~DBConnImp, 63
 - DBConnImp, 63
 - last_auto_increment, 64
 - query, 64
 - select, 64
- gldb::DBConnMySQL, 64
 - ~DBConnMySQL, 66
 - DBConnMySQL, 66
 - last_auto_increment, 66
 - m_conn, 67
 - mtx, 67
 - operator=, 66, 67
 - query, 67
 - select, 67
- gldb::MySQLResult, 100
 - ~MySQLResult, 100
 - m_num_fields, 101
 - m_result, 101
 - MySQLResult, 100, 101
 - num_fields, 101
 - operator=, 101
 - result, 101
- gldb::Table, 104
 - ~Table, 106
 - append_record, 107
 - begin, 107
 - create_from_file, 107
 - end, 107, 108
 - get_field, 108
 - get_headers, 108
 - insert_query, 108
 - m_headers, 110
 - m_quoted, 110
 - m_records, 110
 - num_fields, 109
 - num_records, 109

- operator=, [109](#)
- set_quoted, [110](#)
- Table, [106](#)
- gldb::TableBadInputFile, [110](#)
 - TableBadInputFile, [111](#)
- gldb::TableCouldNotOpenInputFile, [112](#)
 - TableCouldNotOpenInputFile, [112](#)
- gldb::TableException, [113](#)
 - TableException, [113](#)
- gldb::TableField, [114](#)
 - ~TableField, [116](#)
 - length, [116](#)
 - m_data, [119](#)
 - operator std::string, [116](#)
 - operator<<, [118](#)
 - operator+=, [116](#), [117](#)
 - operator=, [117](#), [118](#)
 - TableField, [115](#), [116](#)
- gldb::TableMismatchedRecordLength, [119](#)
 - TableMismatchedRecordLength, [120](#)
- gldb::TableNoSuchField, [120](#)
 - TableNoSuchField, [121](#)
- gldb::TableNoSuchRecord, [122](#)
 - TableNoSuchRecord, [122](#)
- gldb::TableRow, [123](#)
 - ~TableRow, [125](#)
 - append_field, [125](#), [126](#)
 - begin, [126](#)
 - end, [126](#)
 - m_fields, [128](#)
 - operator=, [127](#)
 - print, [127](#)
 - record_string, [128](#)
 - size, [128](#)
 - TableRow, [124](#), [125](#)
- gluser_pass.cpp
 - _XOPEN_SOURCE, [194](#)
 - generate_salt, [194](#)
- grant
 - genleg::DBSQLStatements, [73](#)
 - genleg::GLDatabase, [83](#)
- grow_widths
 - General Ledger database module., [23](#)
- height
 - pgcurses::Size, [104](#)
- id
 - genleg::GLUser, [97](#)
- insert_query
 - gldb::Table, [108](#)
- is_set
 - genleg::Config, [44](#)
- je_by_id
 - genleg::DBSQLStatements, [74](#)
- je_report
 - genleg::GLDatabase, [83](#)
- jelines_by_id
 - genleg::DBSQLStatements, [74](#)
- join
 - General purpose utilities., [27](#)
- last_auto_increment
 - gldb::DBConn, [56](#)
 - gldb::DBConnImp, [64](#)
 - gldb::DBConnMySQL, [66](#)
- lastname
 - genleg::GLUser, [97](#)
- length
 - gldb::TableField, [116](#)
- lib/config/config.cpp, [139](#)
- lib/config/config.h, [140](#)
- lib/config/config_getopt.cpp, [141](#)
- lib/database/data_structures.h, [142](#)
- lib/database/database.h, [143](#)
- lib/database/dbconn.cpp, [145](#)
- lib/database/dbconn.h, [146](#)
- lib/database/dbconnimp.h, [147](#)
- lib/database/table.cpp, [149](#)
- lib/database/table.h, [150](#)
- lib/database/tablefield.cpp, [151](#)
- lib/database/tablefield.h, [152](#)
- lib/database/tablerow.cpp, [153](#)
- lib/database/tablerow.h, [154](#)
- lib/database_imp/database_imp.h, [156](#)
- lib/database_imp/dummy/dbconn_dummy_imp.cpp, [157](#)
- lib/database_imp/dummy/dbconn_dummy_imp.h, [158](#)
- lib/database_imp/mysql/dbconn_mysql_functions.cpp, [160](#)
- lib/database_imp/mysql/dbconn_mysql_imp.cpp, [161](#)
- lib/database_imp/mysql/dbconn_mysql_imp.h, [163](#)
- lib/database_imp/mysql/dbconn_mysql_result.cpp, [164](#)
- lib/database_imp/mysql/dbconn_mysql_result.h, [165](#)
- lib/dbsql/dbsql.h, [166](#)
- lib/dbsql/dbsql_dummy.h, [167](#)
- lib/dbsql/dbsql_functions.h, [169](#)
- lib/dbsql/dbsql_implementations.h, [170](#)
- lib/dbsql/dbsql_mysql.h, [172](#)
- lib/dbsql/dbsqlstatements.cpp, [173](#)
- lib/dbsql/dbsqlstatements.h, [174](#)
- lib/gldb/glaccount.cpp, [175](#)
- lib/gldb/glaccount.h, [175](#)
- lib/gldb/glDATABASE.cpp, [177](#)
- lib/gldb/glDATABASE.h, [178](#)
- lib/gldb/gldb.h, [179](#)
- lib/gldb/glentity.cpp, [180](#)
- lib/gldb/glentity.h, [181](#)
- lib/gldb/glexception.h, [183](#)
- lib/gldb/gljournal.cpp, [184](#)
- lib/gldb/gljournal.h, [184](#)
- lib/gldb/glreport.cpp, [186](#)
- lib/gldb/glreport.h, [187](#)
- lib/gldb/glstanding.cpp, [189](#)
- lib/gldb/glstanding.h, [190](#)
- lib/gldb/gluser.cpp, [191](#)
- lib/gldb/gluser.h, [191](#)
- lib/gldb/gluser_pass.cpp, [193](#)

- lib/pgcurses/pgcurses.h, 194
- lib/pgcurses/termprogram.cpp, 195
- lib/pgcurses/termprogram.h, 196
- lib/pgcurses/termprogramimp.cpp, 197
- lib/pgcurses/termprogramimp.h, 198
- lib/pgcurses/tpexception.h, 199
- lib/pgcurses/tpgentypes.h, 200
- lib/pgcurses/tpwindow.cpp, 201
- lib/pgcurses/tpwindow.h, 202
- lib/pgcurses/tpwindowimp.cpp, 203
- lib/pgcurses/tpwindowimp.h, 204
- lib/pgutils/currency.cpp, 205
- lib/pgutils/currency.h, 205
- lib/pgutils/pgutils.h, 207
- lib/pgutils/stringhelp.cpp, 208
- lib/pgutils/stringhelp.h, 209
- list_users_report
 - genleg::GLDatabase, 83
- listusers
 - genleg::DBSQLStatements, 74
- load_sample_data
 - genleg::GLDatabase, 83
- login
 - Database program., 34
 - Reporting program., 36
 - Terminal access program., 38
 - User administration program., 40
- m_acct
 - genleg::GLJELine, 90
- m_aggregate
 - genleg::GLEntity, 88
- m_amount
 - genleg::GLJELine, 90
- m_commit
 - genleg::GLDBTransaction, 87
- m_conn
 - gldb::DBConnMySQL, 67
- m_data
 - gldb::TableField, 119
- m_dbc
 - genleg::GLDatabase, 84
 - genleg::GLDBTransaction, 87
- m_description
 - genleg::GLAccount, 77
- m_enabled
 - genleg::GLAccount, 77
 - genleg::GLEntity, 88
 - genleg::GLUser, 99
- m_entity
 - genleg::GLJournal, 92
- m_fields
 - gldb::TableRow, 128
- m_firstname
 - genleg::GLUser, 99
- m_frac
 - pgutils::Currency, 54
- m_headers
 - genleg::GLReport, 93
- gldb::Table, 110
- m_id
 - genleg::GLJournal, 92
 - genleg::GLUser, 99
- m_imp
 - gldb::DBConn, 57
 - pgcurses::TermProgram, 130
 - pgcurses::TPWindow, 135
- m_int
 - pgutils::Currency, 54
- m_lastname
 - genleg::GLUser, 99
- m_lines
 - genleg::GLJournal, 92
- m_mainwin
 - pgcurses::TermProgramImp, 132
- m_memo
 - genleg::GLJournal, 92
- m_name
 - genleg::GLEntity, 88
- m_num_fields
 - gldb::MySQLResult, 101
- m_num_periods
 - genleg::GLStandingData, 94
- m_old_cursor
 - pgcurses::TermProgramImp, 132
- m_opts_set
 - genleg::Config, 45
- m_opts_supp
 - genleg::Config, 45
- m_origin
 - pgcurses::TPWindowImp, 138
- m_parent
 - genleg::GLEntity, 88
- m_pass_hash
 - genleg::GLUser, 99
- m_pass_salt
 - genleg::GLUser, 99
- m_period
 - genleg::GLJournal, 92
 - genleg::GLStandingData, 94
- m_perms
 - genleg::GLUser, 99
- m_quoted
 - gldb::Table, 110
- m_records
 - gldb::Table, 110
- m_report_text
 - genleg::GLReport, 93
- m_result
 - gldb::MySQLResult, 101
- m_shortcode
 - genleg::GLEntity, 88
- m_size
 - pgcurses::TPWindowImp, 138
- m_source
 - genleg::GLJournal, 92
- m_sql

- genleg::GLDatabase, 85
- m_tables
 - genleg::GLDatabase, 85
- m_termsize
 - pgcurses::TermProgramImp, 132
- m_title
 - genleg::GLReport, 94
- m_user
 - genleg::GLJournal, 93
- m_username
 - genleg::GLUser, 99
- m_views
 - genleg::GLDatabase, 85
- m_win
 - pgcurses::TPWindowImp, 138
- m_year
 - genleg::GLJournal, 93
 - genleg::GLStandingData, 95
- main
 - Database program., 34
 - Reporting program., 36
 - Terminal access program., 38
 - User administration program., 41
- max_column_widths
 - General Ledger database module., 23
- mtx
 - glldb::DBConnMySQL, 67
- MySQLResult
 - glldb::MySQLResult, 100, 101
- next_content_line
 - General purpose utilities., 27
- num_fields
 - glldb::MySQLResult, 101
 - glldb::Table, 109
- num_records
 - glldb::Table, 109
- operator std::string
 - glldb::TableField, 116
- operator<
 - General purpose utilities., 28
 - pgutils::Currency, 53
- operator<<
 - glldb::TableField, 118
- operator<=
 - General purpose utilities., 29
- operator>
 - General purpose utilities., 29
- operator>=
 - General purpose utilities., 29
- operator+
 - General purpose utilities., 28
 - pgutils::Currency, 53
- operator+=
 - glldb::TableField, 116, 117
 - pgutils::Currency, 52
- operator-
 - General purpose utilities., 28
- pgutils::Currency, 52
- operator==
 - pgutils::Currency, 52
- operator=
 - glldb::DBConn, 56
 - glldb::DBConnDummy, 61
 - glldb::DBConnMySQL, 66, 67
 - glldb::MySQLResult, 101
 - glldb::Table, 109
 - glldb::TableField, 117, 118
 - glldb::TableRow, 127
 - pgcurses::TermProgram, 129
 - pgcurses::TermProgramImp, 131
 - pgcurses::TPWindow, 134
 - pgcurses::TPWindowImp, 137
- operator==
 - General purpose utilities., 29
 - pgutils::Currency, 53
- origin
 - pgcurses::Rectangle, 103
- pass_hash
 - genleg::GLUser, 97
- pass_salt
 - genleg::GLUser, 98
- permissions
 - genleg::GLUser, 98
- pgcurses::Point, 102
 - x, 102
 - y, 102
- pgcurses::Rectangle, 102
 - origin, 103
 - size, 103
- pgcurses::Size, 103
 - height, 104
 - width, 104
- pgcurses::TPEException, 132
 - TPEException, 133
- pgcurses::TPWindow, 133
 - ~TPWindow, 134
 - m_imp, 135
 - operator=, 134
 - TPWindow, 134
 - write_char, 134, 135
 - write_str, 135
- pgcurses::TPWindowImp, 135
 - ~TPWindowImp, 137
 - m_origin, 138
 - m_size, 138
 - m_win, 138
 - operator=, 137
 - TPWindowImp, 137
 - write_char, 137, 138
 - write_str, 138
- pgcurses::TermProgram, 128
 - ~TermProgram, 129
 - m_imp, 130
 - operator=, 129
 - sleep, 129

- TermProgram, 129
- terminal_size, 130
- pgcurses::TermProgramImp, 130
 - ~TermProgramImp, 131
 - m_mainwin, 132
 - m_old_cursor, 132
 - m_termsize, 132
 - operator=, 131
 - sleep, 131
 - TermProgramImp, 131
 - terminal_size, 132
- pgutils::Currency, 51
 - Currency, 52
 - expand, 52
 - m_frac, 54
 - m_int, 54
 - operator<, 53
 - operator+, 53
 - operator+=, 52
 - operator-, 52
 - operator-=, 52
 - operator==, 53
- pgutils::CurrencyException, 54
 - CurrencyException, 54
- plain_report_from_table
 - General Ledger database module., 23
- plain_row
 - General Ledger database module., 24
- populate_from_cmdline
 - genleg::Config, 44
- populate_from_file
 - genleg::Config, 45
- post_je
 - genleg::DBSQLStatements, 74
- post_je_line
 - genleg::DBSQLStatements, 75
- post_journal
 - genleg::GLDatabase, 84
- print
 - gldb::TableRow, 127
- Program configuration module, 17
- progs/gl_db/gl_db_main.cpp, 210
- progs/gl_report/gl_report_main.cpp, 212
- progs/gl_term/gl_term_main.cpp, 213
- progs/gl_user/gl_user_main.cpp, 214
- query
 - gldb::DBConn, 56
 - gldb::DBConnDummy, 61
 - gldb::DBConnImp, 64
 - gldb::DBConnMySQL, 67
- record_string
 - gldb::TableRow, 128
- replace
 - General purpose utilities., 30
- report
 - genleg::GLDatabase, 84
- Reporting program., 35
 - check_db_parameters, 35
 - check_help_and_version, 35
 - login, 36
 - main, 36
 - set_configuration, 36
- result
 - gldb::MySQLResult, 101
- revoke
 - genleg::DBSQLStatements, 75
 - genleg::GLDatabase, 84
- SQL statements module, 21
- select
 - gldb::DBConn, 56
 - gldb::DBConnDummy, 61
 - gldb::DBConnImp, 64
 - gldb::DBConnMySQL, 67
- separator_row
 - General Ledger database module., 24
- set_configuration
 - Database program., 34
 - Reporting program., 36
 - Terminal access program., 38
 - User administration program., 41
- set_enabled
 - genleg::GLUser, 98
- set_firstname
 - genleg::GLUser, 98
- set_lastname
 - genleg::GLUser, 98
- set_password
 - genleg::GLUser, 98
- set_quoted
 - gldb::Table, 110
- set_user_password
 - User administration program., 41
- set_username
 - genleg::GLUser, 99
- show_user_details
 - User administration program., 41
- size
 - gldb::TableRow, 128
 - pgcurses::Rectangle, 103
- sleep
 - pgcurses::TermProgram, 129
 - pgcurses::TermProgramImp, 131
- split
 - General purpose utilities., 30
- split_lines
 - General purpose utilities., 31
- standing_data
 - genleg::DBSQLStatements, 75
- standing_data_report
 - genleg::GLDatabase, 84
- TPEException
 - pgcurses::TPEException, 133
- TPWindow
 - pgcurses::TPWindow, 134

- TPWindowImp
 - pgcurses::TPWindowImp, [137](#)
- Table
 - gldb::Table, [106](#)
- TableBadInputFile
 - gldb::TableBadInputFile, [111](#)
- TableCouldNotOpenInputFile
 - gldb::TableCouldNotOpenInputFile, [112](#)
- TableException
 - gldb::TableException, [113](#)
- TableField
 - gldb::TableField, [115](#), [116](#)
- TableMismatchedRecordLength
 - gldb::TableMismatchedRecordLength, [120](#)
- TableNoSuchField
 - gldb::TableNoSuchField, [121](#)
- TableNoSuchRecord
 - gldb::TableNoSuchRecord, [122](#)
- TableRow
 - gldb::TableRow, [124](#), [125](#)
- TermProgram
 - pgcurses::TermProgram, [129](#)
- TermProgramImp
 - pgcurses::TermProgramImp, [131](#)
- Terminal access program., [37](#)
 - check_db_parameters, [37](#)
 - check_help_and_version, [37](#)
 - login, [38](#)
 - main, [38](#)
 - set_configuration, [38](#)
- Terminal program library, [25](#)
- terminal_size
 - pgcurses::TermProgram, [130](#)
 - pgcurses::TermProgramImp, [132](#)
- trim
 - General purpose utilities., [31](#)
- trim_back
 - General purpose utilities., [31](#)
- trim_front
 - General purpose utilities., [31](#)
- update_user
 - genleg::DBSQLStatements, [75](#)
 - genleg::GLDatabase, [84](#)
- User administration program., [39](#)
 - check_db_parameters, [39](#)
 - check_help_and_version, [40](#)
 - check_user_password, [40](#)
 - enable_user, [40](#)
 - get_user, [40](#)
 - login, [40](#)
 - main, [41](#)
 - set_configuration, [41](#)
 - set_user_password, [41](#)
 - show_user_details, [41](#)
- user_by_id
 - genleg::DBSQLStatements, [75](#)
- user_by_username
 - genleg::DBSQLStatements, [76](#)
- username
 - genleg::GLUser, [99](#)
- width
 - pgcurses::Size, [104](#)
- write_char
 - pgcurses::TPWindow, [134](#), [135](#)
 - pgcurses::TPWindowImp, [137](#), [138](#)
- write_str
 - pgcurses::TPWindow, [135](#)
 - pgcurses::TPWindowImp, [138](#)
- x
 - pgcurses::Point, [102](#)
- y
 - pgcurses::Point, [102](#)