

general\_ledger

Generated by Doxygen 1.8.1.2

Sat Jun 14 2014 17:27:25



# Contents

<b>1</b>	<b>General Ledger.</b>	<b>1</b>
<b>2</b>	<b>Module Index</b>	<b>3</b>
2.1	Modules . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Class Index</b>	<b>7</b>
4.1	Class List . . . . .	7
<b>5</b>	<b>File Index</b>	<b>9</b>
5.1	File List . . . . .	9
<b>6</b>	<b>Module Documentation</b>	<b>11</b>
6.1	General Ledger database module. . . . .	11
6.1.1	Detailed Description . . . . .	11
6.2	Database interaction module . . . . .	12
6.2.1	Detailed Description . . . . .	12
6.2.2	Function Documentation . . . . .	13
6.2.2.1	get_connection . . . . .	13
6.2.2.2	get_database_type . . . . .	13
6.3	SQL statements module . . . . .	14
6.3.1	Detailed Description . . . . .	14
6.4	Program configuration module . . . . .	15
6.4.1	Detailed Description . . . . .	15
6.5	General purpose helpers. . . . .	16
6.5.1	Detailed Description . . . . .	16
6.5.2	Function Documentation . . . . .	16
6.5.2.1	split . . . . .	16
6.5.2.2	split . . . . .	16
6.5.2.3	trim . . . . .	16
6.5.2.4	trim_back . . . . .	17

6.5.2.5	trim_front . . . . .	17
6.6	Reporting program. . . . .	18
6.6.1	Detailed Description . . . . .	18
6.6.2	Function Documentation . . . . .	18
6.6.2.1	login . . . . .	18
6.6.2.2	main . . . . .	18
6.6.2.3	set_configuration . . . . .	19
6.7	Database program. . . . .	20
6.7.1	Detailed Description . . . . .	20
6.7.2	Function Documentation . . . . .	20
6.7.2.1	check_db_parameters . . . . .	20
6.7.2.2	check_help_and_version . . . . .	20
6.7.2.3	login . . . . .	21
6.7.2.4	main . . . . .	21
6.7.2.5	set_configuration . . . . .	21
<b>7</b>	<b>Class Documentation</b>	<b>23</b>
7.1	genleg::Config Class Reference . . . . .	23
7.1.1	Detailed Description . . . . .	23
7.1.2	Constructor & Destructor Documentation . . . . .	23
7.1.2.1	Config . . . . .	23
7.1.2.2	~Config . . . . .	24
7.1.3	Member Function Documentation . . . . .	24
7.1.3.1	add_cmdline_option . . . . .	24
7.1.3.2	is_set . . . . .	24
7.1.3.3	operator[] . . . . .	24
7.1.3.4	populate_from_cmdline . . . . .	24
7.1.3.5	populate_from_file . . . . .	25
7.1.4	Member Data Documentation . . . . .	25
7.1.4.1	m_opts_set . . . . .	25
7.1.4.2	m_opts_supp . . . . .	25
7.2	genleg::ConfigBadConfigFile Class Reference . . . . .	25
7.2.1	Detailed Description . . . . .	26
7.2.2	Constructor & Destructor Documentation . . . . .	26
7.2.2.1	ConfigBadConfigFile . . . . .	26
7.3	genleg::ConfigBadOption Class Reference . . . . .	27
7.3.1	Detailed Description . . . . .	27
7.3.2	Constructor & Destructor Documentation . . . . .	27
7.3.2.1	ConfigBadOption . . . . .	28
7.4	genleg::ConfigCouldNotOpenFile Class Reference . . . . .	28

7.4.1	Detailed Description	29
7.4.2	Constructor & Destructor Documentation	29
7.4.2.1	ConfigCouldNotOpenFile	29
7.5	genleg::ConfigException Class Reference	29
7.5.1	Detailed Description	29
7.5.2	Constructor & Destructor Documentation	30
7.5.2.1	ConfigException	30
7.6	genleg::ConfigOptionNotSet Class Reference	30
7.6.1	Detailed Description	31
7.6.2	Constructor & Destructor Documentation	31
7.6.2.1	ConfigOptionNotSet	31
7.7	gldb::DBConn Class Reference	31
7.7.1	Detailed Description	32
7.7.2	Constructor & Destructor Documentation	32
7.7.2.1	DBConn	32
7.7.2.2	DBConn	32
7.7.3	Member Function Documentation	32
7.7.3.1	operator=	32
7.7.3.2	query	32
7.7.3.3	select	32
7.7.4	Member Data Documentation	33
7.7.4.1	m_imp	33
7.8	gldb::DBConnCouldNotConnect Class Reference	33
7.8.1	Detailed Description	34
7.8.2	Constructor & Destructor Documentation	34
7.8.2.1	DBConnCouldNotConnect	34
7.9	gldb::DBConnCouldNotQuery Class Reference	34
7.9.1	Detailed Description	35
7.9.2	Constructor & Destructor Documentation	35
7.9.2.1	DBConnCouldNotQuery	35
7.10	gldb::DBConnDummy Class Reference	35
7.10.1	Detailed Description	36
7.10.2	Constructor & Destructor Documentation	36
7.10.2.1	DBConnDummy	36
7.10.2.2	DBConnDummy	37
7.10.2.3	~DBConnDummy	37
7.10.3	Member Function Documentation	37
7.10.3.1	operator=	37
7.10.3.2	select	37
7.11	gldb::DBConnException Class Reference	37

7.11.1 Detailed Description . . . . .	38
7.11.2 Constructor & Destructor Documentation . . . . .	38
7.11.2.1 DBConnException . . . . .	38
7.12 glDb::DBConnImp Class Reference . . . . .	38
7.12.1 Detailed Description . . . . .	39
7.12.2 Constructor & Destructor Documentation . . . . .	39
7.12.2.1 DBConnImp . . . . .	39
7.12.2.2 ~DBConnImp . . . . .	39
7.12.3 Member Function Documentation . . . . .	39
7.12.3.1 query . . . . .	39
7.12.3.2 select . . . . .	40
7.13 glDb::DBConnMySQL Class Reference . . . . .	40
7.13.1 Detailed Description . . . . .	41
7.13.2 Constructor & Destructor Documentation . . . . .	41
7.13.2.1 DBConnMySQL . . . . .	41
7.13.2.2 DBConnMySQL . . . . .	41
7.13.2.3 ~DBConnMySQL . . . . .	41
7.13.3 Member Function Documentation . . . . .	41
7.13.3.1 operator= . . . . .	42
7.13.3.2 query . . . . .	42
7.13.3.3 select . . . . .	42
7.13.4 Member Data Documentation . . . . .	42
7.13.4.1 m_conn . . . . .	42
7.14 genleg::DBSQLMySQL Class Reference . . . . .	42
7.14.1 Detailed Description . . . . .	43
7.15 genleg::DBSQLStatements Class Reference . . . . .	43
7.15.1 Detailed Description . . . . .	44
7.15.2 Constructor & Destructor Documentation . . . . .	44
7.15.2.1 DBSQLStatements . . . . .	44
7.15.2.2 ~DBSQLStatements . . . . .	44
7.15.3 Member Function Documentation . . . . .	45
7.15.3.1 create_table . . . . .	45
7.15.3.2 create_view . . . . .	45
7.15.3.3 drop_table . . . . .	45
7.15.3.4 drop_view . . . . .	45
7.15.3.5 user_by_id . . . . .	45
7.15.3.6 user_by_username . . . . .	46
7.16 genleg::GLDatabase Class Reference . . . . .	46
7.16.1 Detailed Description . . . . .	47
7.16.2 Constructor & Destructor Documentation . . . . .	47

7.16.2.1	GLDatabase	47
7.16.2.2	~GLDatabase	48
7.16.3	Member Function Documentation	48
7.16.3.1	backend	48
7.16.3.2	create_structure	48
7.16.3.3	destroy_structure	48
7.16.3.4	get_user_by_id	48
7.16.3.5	get_user_by_username	48
7.16.3.6	load_sample_data	49
7.16.4	Member Data Documentation	49
7.16.4.1	m_dbc	49
7.16.4.2	m_sql	49
7.16.4.3	m_tables	49
7.16.4.4	m_views	49
7.17	genleg::GLDBException Class Reference	49
7.17.1	Detailed Description	50
7.17.2	Constructor & Destructor Documentation	50
7.17.2.1	GLDBException	50
7.18	genleg::GLUser Class Reference	50
7.18.1	Detailed Description	51
7.18.2	Constructor & Destructor Documentation	51
7.18.2.1	GLUser	51
7.18.2.2	~GLUser	51
7.18.3	Member Function Documentation	51
7.18.3.1	enabled	51
7.18.3.2	firstname	52
7.18.3.3	id	52
7.18.3.4	lastname	52
7.18.3.5	username	52
7.18.4	Member Data Documentation	52
7.18.4.1	m_enabled	52
7.18.4.2	m_firstname	52
7.18.4.3	m_id	52
7.18.4.4	m_lastname	52
7.18.4.5	m_username	53
7.19	gldb::Table Class Reference	53
7.19.1	Detailed Description	54
7.19.2	Constructor & Destructor Documentation	54
7.19.2.1	Table	54
7.19.2.2	~Table	54

7.19.3	Member Function Documentation	54
7.19.3.1	append_record	54
7.19.3.2	create_from_file	54
7.19.3.3	get_field	55
7.19.3.4	get_headers	55
7.19.3.5	insert_query	55
7.19.3.6	num_fields	55
7.19.3.7	num_records	56
7.19.3.8	operator[]	56
7.19.3.9	set_quoted	56
7.19.4	Member Data Documentation	56
7.19.4.1	m_headers	56
7.19.4.2	m_quoted	56
7.19.4.3	m_records	56
7.20	gldb::TableBadInputFile Class Reference	56
7.20.1	Detailed Description	57
7.20.2	Constructor & Destructor Documentation	57
7.20.2.1	TableBadInputFile	57
7.21	gldb::TableCouldNotOpenInputFile Class Reference	58
7.21.1	Detailed Description	58
7.21.2	Constructor & Destructor Documentation	58
7.21.2.1	TableCouldNotOpenInputFile	59
7.22	gldb::TableException Class Reference	59
7.22.1	Detailed Description	59
7.22.2	Constructor & Destructor Documentation	59
7.22.2.1	TableException	60
7.23	gldb::TableField Class Reference	60
7.23.1	Detailed Description	61
7.23.2	Constructor & Destructor Documentation	61
7.23.2.1	TableField	61
7.23.2.2	TableField	61
7.23.2.3	~TableField	61
7.23.3	Member Function Documentation	61
7.23.3.1	length	61
7.23.3.2	operator std::string	62
7.23.3.3	operator+=	62
7.23.3.4	operator+=	62
7.23.3.5	operator=	62
7.23.3.6	operator=	62
7.23.3.7	operator[]	63



7.23.3.8	operator[]	63
7.23.4	Friends And Related Function Documentation	63
7.23.4.1	operator<<	63
7.23.5	Member Data Documentation	63
7.23.5.1	m_data	63
7.24	gldb::TableMismatchedRecordLength Class Reference	63
7.24.1	Detailed Description	64
7.24.2	Constructor & Destructor Documentation	64
7.24.2.1	TableMismatchedRecordLength	64
7.25	gldb::TableNoSuchField Class Reference	65
7.25.1	Detailed Description	65
7.25.2	Constructor & Destructor Documentation	66
7.25.2.1	TableNoSuchField	66
7.26	gldb::TableNoSuchRecord Class Reference	66
7.26.1	Detailed Description	67
7.26.2	Constructor & Destructor Documentation	67
7.26.2.1	TableNoSuchRecord	67
7.27	gldb::TableRow Class Reference	67
7.27.1	Detailed Description	68
7.27.2	Constructor & Destructor Documentation	68
7.27.2.1	TableRow	68
7.27.2.2	TableRow	68
7.27.2.3	TableRow	68
7.27.2.4	~TableRow	68
7.27.3	Member Function Documentation	68
7.27.3.1	append_field	68
7.27.3.2	append_field	68
7.27.3.3	append_field	69
7.27.3.4	operator[]	69
7.27.3.5	operator[]	69
7.27.3.6	print	69
7.27.3.7	record_string	69
7.27.3.8	record_string	70
7.27.3.9	size	70
7.27.4	Member Data Documentation	70
7.27.4.1	m_fields	70
<b>8</b>	<b>File Documentation</b>	<b>71</b>
8.1	lib/config/config.cpp File Reference	71
8.1.1	Detailed Description	71

8.2	lib/config/config.h File Reference	72
8.2.1	Detailed Description	73
8.3	lib/config/config_getopt.cpp File Reference	73
8.3.1	Detailed Description	73
8.3.2	Macro Definition Documentation	74
8.3.2.1	_XOPEN_SOURCE	74
8.4	lib/database/data_structures.h File Reference	74
8.4.1	Detailed Description	75
8.5	lib/database/database.h File Reference	75
8.5.1	Detailed Description	77
8.6	lib/database/dbconn.cpp File Reference	77
8.6.1	Detailed Description	77
8.7	lib/database/dbconn.h File Reference	78
8.7.1	Detailed Description	79
8.8	lib/database/dbconnimp.h File Reference	79
8.8.1	Detailed Description	81
8.9	lib/database/table.cpp File Reference	81
8.9.1	Detailed Description	81
8.10	lib/database/table.h File Reference	82
8.10.1	Detailed Description	83
8.11	lib/database/tablefield.cpp File Reference	84
8.11.1	Detailed Description	84
8.12	lib/database/tablefield.h File Reference	84
8.12.1	Detailed Description	86
8.13	lib/database/ablerow.cpp File Reference	86
8.13.1	Detailed Description	86
8.14	lib/database/ablerow.h File Reference	87
8.14.1	Detailed Description	88
8.15	lib/database_imp/database_imp.h File Reference	88
8.15.1	Detailed Description	90
8.16	lib/database_imp/dummy/dbconn_dummy_imp.cpp File Reference	90
8.16.1	Detailed Description	91
8.17	lib/database_imp/dummy/dbconn_dummy_imp.h File Reference	91
8.17.1	Detailed Description	93
8.18	lib/database_imp/mysql/dbconn_mysql_imp.cpp File Reference	93
8.18.1	Detailed Description	94
8.19	lib/database_imp/mysql/dbconn_mysql_imp.h File Reference	94
8.19.1	Detailed Description	96
8.20	lib/dbsql/dbsql_mysql.h File Reference	96
8.20.1	Detailed Description	97

8.21	<a href="#">lib/dbsql/dbsqlstatements.cpp File Reference</a>	98
8.21.1	Detailed Description	98
8.22	<a href="#">lib/dbsql/dbsqlstatements.h File Reference</a>	99
8.22.1	Detailed Description	100
8.23	<a href="#">lib/gldb/gldatabase.cpp File Reference</a>	101
8.23.1	Detailed Description	101
8.24	<a href="#">lib/gldb/gldatabase.h File Reference</a>	101
8.24.1	Detailed Description	103
8.25	<a href="#">lib/gldb/gldb.h File Reference</a>	103
8.25.1	Detailed Description	104
8.26	<a href="#">lib/gldb/glexception.h File Reference</a>	104
8.26.1	Detailed Description	105
8.27	<a href="#">lib/gldb/gluser.cpp File Reference</a>	106
8.27.1	Detailed Description	106
8.28	<a href="#">lib/gldb/gluser.h File Reference</a>	106
8.28.1	Detailed Description	107
8.29	<a href="#">lib/stringhelp/stringhelp.cpp File Reference</a>	108
8.29.1	Detailed Description	108
8.30	<a href="#">lib/stringhelp/stringhelp.h File Reference</a>	108
8.30.1	Detailed Description	110
8.31	<a href="#">progs/gl_db/gl_db_main.cpp File Reference</a>	110
8.31.1	Detailed Description	111
8.32	<a href="#">progs/gl_report/gl_report_main.cpp File Reference</a>	111
8.32.1	Detailed Description	113
8.33	<a href="#">progs/gl_user/gl_user_main.cpp File Reference</a>	113
8.33.1	Detailed Description	114
8.33.2	Function Documentation	114
8.33.2.1	<a href="#">check_db_parameters</a>	114
8.33.2.2	<a href="#">check_help_and_version</a>	114
8.33.2.3	<a href="#">login</a>	115
8.33.2.4	<a href="#">main</a>	115
8.33.2.5	<a href="#">set_configuration</a>	115



## 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

# Module Index

### 2.1 Modules

Here is a list of all modules:

General Ledger database module. . . . .	11
Database interaction module . . . . .	12
SQL statements module . . . . .	14
Program configuration module . . . . .	15
General purpose helpers. . . . .	16
Reporting program. . . . .	18
Database program. . . . .	20





## Chapter 3

# Class Index

### 3.1 Class Hierarchy

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

genleg::Config . . . . .	23
genleg::ConfigException . . . . .	29
genleg::ConfigBadConfigFile . . . . .	25
genleg::ConfigBadOption . . . . .	27
genleg::ConfigCouldNotOpenFile . . . . .	28
genleg::ConfigOptionNotSet . . . . .	30
gldb::DBConn . . . . .	31
gldb::DBConnException . . . . .	37
gldb::DBConnCouldNotConnect . . . . .	33
gldb::DBConnCouldNotQuery . . . . .	34
gldb::DBConnImp . . . . .	38
gldb::DBConnDummy . . . . .	35
gldb::DBConnMySQL . . . . .	40
genleg::DBSQLStatements . . . . .	43
genleg::DBSQLMySQL . . . . .	42
genleg::GLDatabase . . . . .	46
genleg::GLDBException . . . . .	49
genleg::GLUser . . . . .	50
gldb::Table . . . . .	53
gldb::TableException . . . . .	59
gldb::TableBadInputFile . . . . .	56
gldb::TableCouldNotOpenInputFile . . . . .	58
gldb::TableMismatchedRecordLength . . . . .	63
gldb::TableNoSuchField . . . . .	65
gldb::TableNoSuchRecord . . . . .	66
gldb::TableField . . . . .	60
gldb::TableRow . . . . .	67



## Chapter 4

# Class Index

### 4.1 Class List

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

<a href="#">genleg::Config</a>	Configuration options class . . . . .	23
<a href="#">genleg::ConfigBadConfigFile</a>	Exception class for badly formed configuration file . . . . .	25
<a href="#">genleg::ConfigBadOption</a>	Exception class for bad provided option . . . . .	27
<a href="#">genleg::ConfigCouldNotOpenFile</a>	Exception class for when conf file cannot be opened . . . . .	28
<a href="#">genleg::ConfigException</a>	Configuration module exception base class . . . . .	29
<a href="#">genleg::ConfigOptionNotSet</a>	Exception class for option not set . . . . .	30
<a href="#">gldb::DBConn</a>	Database connection class . . . . .	31
<a href="#">gldb::DBConnCouldNotConnect</a>	Could not connect to database exception class . . . . .	33
<a href="#">gldb::DBConnCouldNotQuery</a>	Could not execute database query exception class . . . . .	34
<a href="#">gldb::DBConnDummy</a>	Dummy database implementation class . . . . .	35
<a href="#">gldb::DBConnException</a>	Base database connection exception class . . . . .	37
<a href="#">gldb::DBConnImp</a>	Abstract database implementation base class . . . . .	38
<a href="#">gldb::DBConnMySQL</a>	MySQL database implementation class . . . . .	40
<a href="#">genleg::DBSQLMySQL</a>	MySQL SQL statements class . . . . .	42
<a href="#">genleg::DBSQLStatements</a>	SQL statements class . . . . .	43
<a href="#">genleg::GLDatabase</a>	General ledger database class . . . . .	46
<a href="#">genleg::GLDBException</a>	Base general ledger database exceptionc class . . . . .	49
<a href="#">genleg::GLUser</a>	General ledger user class . . . . .	50
<a href="#">gldb::Table</a>	Database table class . . . . .	53

<a href="#">gldb::TableBadInputFile</a>	
Could not connect to database exception class . . . . .	56
<a href="#">gldb::TableCouldNotOpenInputFile</a>	
Could not connect to database exception class . . . . .	58
<a href="#">gldb::TableException</a>	
Base database connection exception class . . . . .	59
<a href="#">gldb::TableField</a>	
Database table field class . . . . .	60
<a href="#">gldb::TableMismatchedRecordLength</a>	
Mismatched record length exception class . . . . .	63
<a href="#">gldb::TableNoSuchField</a>	
No such field exception class . . . . .	65
<a href="#">gldb::TableNoSuchRecord</a>	
No such record exception class . . . . .	66
<a href="#">gldb::TableRow</a>	
Database table row class . . . . .	67

## Chapter 5

# File Index

### 5.1 File List

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

lib/config/ <a href="#">config.cpp</a>	
Implementation of program configurations class . . . . .	71
lib/config/ <a href="#">config.h</a>	
Interface to program configurations class . . . . .	72
lib/config/ <a href="#">config_getopt.cpp</a>	
Implementation of command line functionality . . . . .	73
lib/database/ <a href="#">data_structures.h</a>	
Main interface to database data structures . . . . .	74
lib/database/ <a href="#">database.h</a>	
User interface to database functionality . . . . .	75
lib/database/ <a href="#">dbconn.cpp</a>	
Implementation of database connection class . . . . .	77
lib/database/ <a href="#">dbconn.h</a>	
Interface to database connection base class . . . . .	78
lib/database/ <a href="#">dbconnimp.h</a>	
Interface to abstract database implementation base class . . . . .	79
lib/database/ <a href="#">table.cpp</a>	
Implementation of database table data structure . . . . .	81
lib/database/ <a href="#">table.h</a>	
Interface to database table data structure . . . . .	82
lib/database/ <a href="#">tablefield.cpp</a>	
Implementation of database table field class . . . . .	84
lib/database/ <a href="#">tablefield.h</a>	
Interface to database table field class . . . . .	84
lib/database/ <a href="#">tablerow.cpp</a>	
Implementation of database table row data structure . . . . .	86
lib/database/ <a href="#">tablerow.h</a>	
Interface to database table row data structure . . . . .	87
lib/database_imp/ <a href="#">database_imp.h</a>	
Interface to database implementation factory function . . . . .	88
lib/database_imp/dummy/ <a href="#">dbconn_dummy_imp.cpp</a>	
Implementation of Dummy database connection implementation class . . . . .	90
lib/database_imp/dummy/ <a href="#">dbconn_dummy_imp.h</a>	
Interface to dummy database connection implementation class . . . . .	91
lib/database_imp/mysql/ <a href="#">dbconn_mysql_imp.cpp</a>	
Implementation of MySQL database connection implementation class . . . . .	93
lib/database_imp/mysql/ <a href="#">dbconn_mysql_imp.h</a>	
Interface to MySQL database connection implementation class . . . . .	94

lib/dbsql/ <b>dbsql.h</b>	??
lib/dbsql/ <b>dbsql_functions.h</b>	??
lib/dbsql/ <b>dbsql_implementations.h</b>	??
lib/dbsql/ <b>dbsql_mysql.h</b>	
Interface to MySQL SQL statement class	96
lib/dbsql/ <b>dbsqlstatements.cpp</b>	
Implementation of SQL statement class	98
lib/dbsql/ <b>dbsqlstatements.h</b>	
Interface to SQL statement class	99
lib/gldb/ <b>gldatabase.cpp</b>	
Implementation of General Ledger database class	101
lib/gldb/ <b>gldatabase.h</b>	
Interface to General Ledger database class	101
lib/gldb/ <b>gldb.h</b>	
User interface to General Ledger database module	103
lib/gldb/ <b>glexception.h</b>	
Interface to General Ledger base exception class	104
lib/gldb/ <b>gluser.cpp</b>	
Implementation of user class	106
lib/gldb/ <b>gluser.h</b>	
Interface to user class	106
lib/stringhelp/ <b>stringhelp.cpp</b>	
Implementation of string helper functions	108
lib/stringhelp/ <b>stringhelp.h</b>	
Interface to string helper functions	108
progs/gl_db/ <b>gl_db_main.cpp</b>	
Main functionality for gl_db program	110
progs/gl_report/ <b>gl_report_main.cpp</b>	
Main functionality for gl_report program	111
progs/gl_user/ <b>gl_user_main.cpp</b>	
Main functionality for gl_user program	113

## Chapter 6

# Module Documentation

### 6.1 General Ledger database module.

#### Classes

- class [genleg::GLDatabase](#)  
*General ledger database class.*
- class [genleg::GLDBException](#)  
*Base general ledger database exceptionc class.*
- class [genleg::GLUser](#)  
*General ledger user class.*

#### 6.1.1 Detailed Description

Module for interacting with the general ledger database model.

## 6.2 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.*

### 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.*

#### 6.2.1 Detailed Description

Module for interacting with the database.



## 6.2.2 Function Documentation

### 6.2.2.1 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.

### 6.2.2.2 std::string glldb::get\_database\_type ( )

Returns the name of the compiled-in database type.

#### Returns

The name of the compiled-in database type.

## 6.3 SQL statements module

### Classes

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

### 6.3.1 Detailed Description

Module for producing SQL statements used by program.

## 6.4 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.*

#### 6.4.1 Detailed Description

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

## 6.5 General purpose helpers.

### Functions

- `std::string & pgstring::trim_front (std::string &s)`  
*Trims leading whitespace from a string.*
- `std::string & pgstring::trim_back (std::string &s)`  
*Trims trailing whitespace from a string.*
- `std::string & pgstring::trim (std::string &s)`  
*Trims leading and trailing whitespace from a string.*
- `std::vector< std::string > pgstring::split (const std::string &s, const char delim)`  
*Splits a delimited string into tokens.*
- `std::vector< std::string > & pgstring::split (std::vector< std::string > &vec, const std::string &s, const char delim)`  
*Splits a delimited string into tokens.*

#### 6.5.1 Detailed Description

General purpose helper classes and functions.

#### 6.5.2 Function Documentation

##### 6.5.2.1 `std::vector< std::string > pgstring::split ( const std::string & s, const char delim )`

Splits a delimited string into tokens.

###### Parameters

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

###### Returns

A vector of tokens.

##### 6.5.2.2 `std::vector< std::string > & pgstring::split ( std::vector< std::string > & vec, const std::string & s, const char delim )`

Splits a delimited string into tokens.

###### Parameters

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

###### Returns

A reference to `vec`.

##### 6.5.2.3 `std::string & pgstring::trim ( std::string & s )`

Trims leading and trailing whitespace from a string.

## Parameters

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

## Returns

The trimmed string.

**6.5.2.4** `std::string & pgstring::trim_back ( std::string & s )`

Trims trailing whitespace from a string.

## Parameters

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

## Returns

The trimmed string.

**6.5.2.5** `std::string & pgstring::trim_front ( std::string & s )`

Trims leading whitespace from a string.

## Parameters

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

## Returns

The trimmed string.

## 6.6 Reporting program.

### Functions

- static void `set_configuration` (`genleg::Config` &config, int argc, char \*argv[])  
*Sets program configuration options.*
- 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\_report"  
*Static variable for program name.*

#### 6.6.1 Detailed Description

Administrative reporting program.

#### 6.6.2 Function Documentation

##### 6.6.2.1 static std::string login ( void ) [static]

Gets a password from the terminal.

##### Returns

The password.

##### 6.6.2.2 int main ( int argc, char \* argv[] )

Main function.

##### Parameters

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

#### Returns

Exit status code.

6.6.2.3 `static void set_configuration ( genleg::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> .

## 6.7 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 \* `progrname` = "gl\_db"  
*Static variable for program name.*

#### 6.7.1 Detailed Description

Administrative database management program.

#### 6.7.2 Function Documentation

##### 6.7.2.1 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.

##### 6.7.2.2 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.

**6.7.2.3 static std::string login ( void ) [static]**

Gets a password from the terminal.

**Returns**

The password.

**6.7.2.4 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.

**6.7.2.5 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> .



## Chapter 7

# Class Documentation

### 7.1 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](#)

#### 7.1.1 Detailed Description

Configuration options class.

#### 7.1.2 Constructor & Destructor Documentation

##### 7.1.2.1 Config::Config ( )

Constructor

### 7.1.2.2 Config::~~Config ( )

Destructor

## 7.1.3 Member Function Documentation

### 7.1.3.1 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.

### 7.1.3.2 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.

### 7.1.3.3 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

<a href="#"><i>ConfigOptionNotSet</i></a>	If the named option has not been set.
---	---------------------------------------

### 7.1.3.4 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

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

## 7.1.3.5 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

<a href="#"><i>ConfigCouldNotOpenFile</i></a>	If the configuration file cannot be opened.
<a href="#"><i>ConfigBadConfigFile</i></a>	If the configuration file is badly formed.

## 7.1.4 Member Data Documentation

## 7.1.4.1 std::map&lt;std::string, std::string&gt; genleg::Config::m\_opts\_set [private]

Map of options which have been set

## 7.1.4.2 std::list&lt;std::pair&lt;std::string, enum Argument&gt; &gt; genleg::Config::m\_opts\_supp [private]

List of options which are supported

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

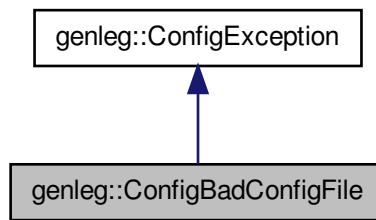
- lib/config/[config.h](#)
- lib/config/[config.cpp](#)
- lib/config/[config\\_getopt.cpp](#)

## 7.2 genleg::ConfigBadConfigFile Class Reference

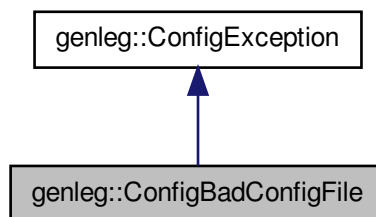
Exception class for badly formed configuration file.

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

Inheritance diagram for `genleg::ConfigBadConfigFile`:



Collaboration diagram for `genleg::ConfigBadConfigFile`:



## Public Member Functions

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

### 7.2.1 Detailed Description

Exception class for badly formed configuration file.

### 7.2.2 Constructor & Destructor Documentation

**7.2.2.1** `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:

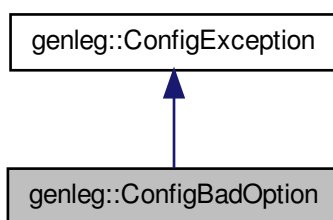
- [lib/config/config.h](#)

## 7.3 genleg::ConfigBadOption Class Reference

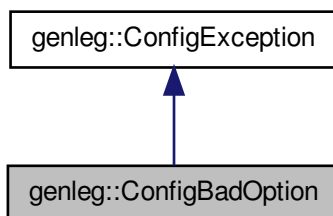
Exception class for bad provided option.

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

Inheritance diagram for genleg::ConfigBadOption:



Collaboration diagram for genleg::ConfigBadOption:



### Public Member Functions

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

#### 7.3.1 Detailed Description

Exception class for bad provided option.

#### 7.3.2 Constructor & Destructor Documentation

### 7.3.2.1 `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:

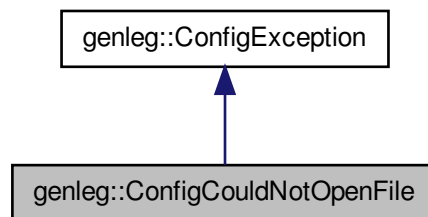
- [lib/config/config.h](#)

## 7.4 `genleg::ConfigCouldNotOpenFile` Class Reference

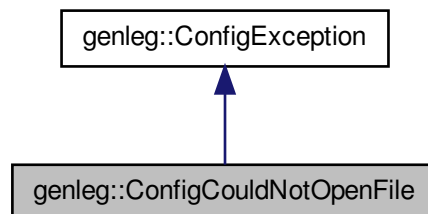
Exception class for when conf file cannot be opened.

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

Inheritance diagram for `genleg::ConfigCouldNotOpenFile`:



Collaboration diagram for `genleg::ConfigCouldNotOpenFile`:



### Public Member Functions

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



### 7.4.1 Detailed Description

Exception class for when conf file cannot be opened.

### 7.4.2 Constructor & Destructor Documentation

**7.4.2.1** `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:

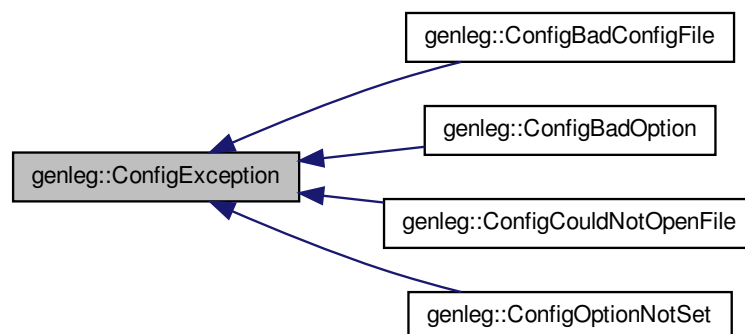
- [lib/config/config.h](#)

## 7.5 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.*

### 7.5.1 Detailed Description

Configuration module exception base class.

## 7.5.2 Constructor & Destructor Documentation

### 7.5.2.1 `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:

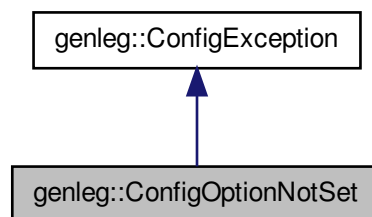
- `lib/config/config.h`

## 7.6 `genleg::ConfigOptionNotSet` Class Reference

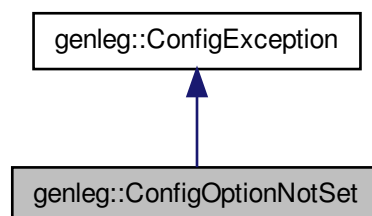
Exception class for option not set.

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

Inheritance diagram for `genleg::ConfigOptionNotSet`:



Collaboration diagram for `genleg::ConfigOptionNotSet`:



## Public Member Functions

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

*Constructor.*

### 7.6.1 Detailed Description

Exception class for option not set.

### 7.6.2 Constructor & Destructor Documentation

#### 7.6.2.1 genleg::ConfigOptionNotSet::ConfigOptionNotSet ( 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:

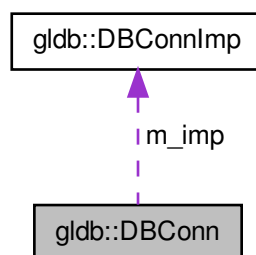
- lib/config/[config.h](#)

## 7.7 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](#) (std::string sql\_query)  
*Runs an SQL query.*
- [Table select](#) (std::string [query](#))

*Runs an SQL SELECT query.*

- [DBConn](#) (const [DBConn](#) &)
- [DBConn](#) & `operator=` (const [DBConn](#) &)

## Private Attributes

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

### 7.7.1 Detailed Description

Database connection class.

### 7.7.2 Constructor & Destructor Documentation

#### 7.7.2.1 `DBConn::DBConn ( DBConnImp * imp )` `[explicit]`

Constructor.

##### Parameters

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

#### 7.7.2.2 `gldb::DBConn::DBConn ( const DBConn & )`

Deleted copy constructor

### 7.7.3 Member Function Documentation

#### 7.7.3.1 `DBConn& gldb::DBConn::operator= ( const DBConn & )`

Deleted assignment operator

#### 7.7.3.2 `void DBConn::query ( std::string sql_query )`

Runs an SQL query.

##### Parameters

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

##### Returns

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

#### 7.7.3.3 `Table DBConn::select ( std::string query )`

Runs an SQL SELECT query.

##### Parameters

<code>query</code>	The query.
--------------------	------------

### Returns

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

## 7.7.4 Member Data Documentation

### 7.7.4.1 DBConnImp\* glldb::DBConn::m\_imp [private]

Pointer to database implementation object.

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

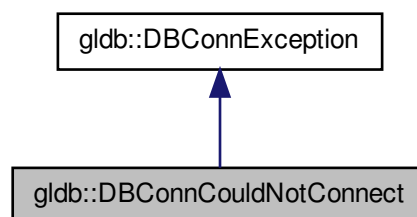
- [lib/database/dbconn.h](#)
- [lib/database/dbconn.cpp](#)

## 7.8 glldb::DBConnCouldNotConnect Class Reference

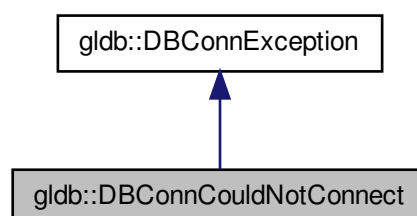
Could not connect to database exception class.

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

Inheritance diagram for glldb::DBConnCouldNotConnect:



Collaboration diagram for glldb::DBConnCouldNotConnect:



## Public Member Functions

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

### 7.8.1 Detailed Description

Could not connect to database exception class.

### 7.8.2 Constructor & Destructor Documentation

7.8.2.1 `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:

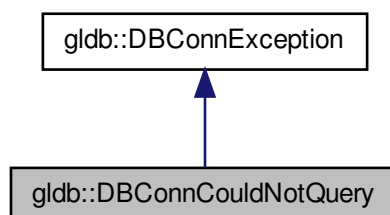
- lib/database/[dbconn.h](#)

## 7.9 gldb::DBConnCouldNotQuery Class Reference

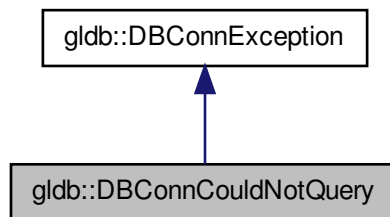
Could not execute database query exception class.

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

Inheritance diagram for gldb::DBConnCouldNotQuery:



Collaboration diagram for glldb::DBConnCouldNotQuery:



## Public Member Functions

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

### 7.9.1 Detailed Description

Could not execute database query exception class.

### 7.9.2 Constructor & Destructor Documentation

7.9.2.1 `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:

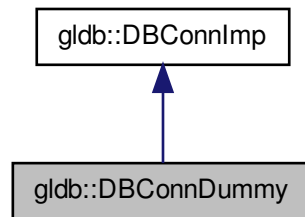
- lib/database/[dbconn.h](#)

## 7.10 glldb::DBConnDummy Class Reference

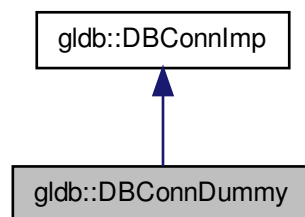
Dummy database implementation class.

```
#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](#) &)
- [Table select](#) (std::string query)  
*Fakes running of an SQL SELECT query.*

### 7.10.1 Detailed Description

Dummy database implementation class.

### 7.10.2 Constructor & Destructor Documentation

- 7.10.2.1 `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.

## 7.10.2.2 gldb::DBConnDummy::DBConnDummy ( const DBConnDummy &amp; )

Deleted copy constructor

## 7.10.2.3 DBConnDummy::~~DBConnDummy ( ) [virtual]

Destructor

## 7.10.3 Member Function Documentation

## 7.10.3.1 DBConnDummy&amp; gldb::DBConnDummy::operator= ( const DBConnDummy &amp; )

Deleted assignment operator

## 7.10.3.2 Table DBConnDummy::select ( 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 [gldb::DBConnImp](#).

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

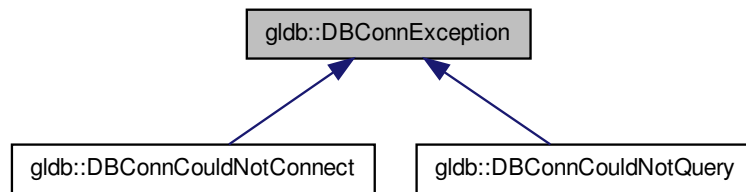
- lib/database\_imp/dummy/dbconn\_dummy\_imp.h
- lib/database\_imp/dummy/dbconn\_dummy\_imp.cpp

## 7.11 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.*

### 7.11.1 Detailed Description

Base database connection exception class.

### 7.11.2 Constructor & Destructor Documentation

7.11.2.1 `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:

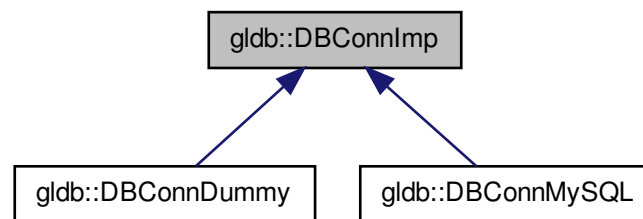
- `lib/database/dbconn.h`

## 7.12 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](#) (std::string sql\_query)=0  
*Runs an SQL query.*
- virtual [Table select](#) (std::string query)=0  
*Runs an SQL SELECT query.*

### 7.12.1 Detailed Description

Abstract database implementation base class.

### 7.12.2 Constructor & Destructor Documentation

7.12.2.1 `glldb::DBConnImp::DBConnImp ( ) [inline]`

Constructor

7.12.2.2 `virtual glldb::DBConnImp::~~DBConnImp ( ) [inline],[virtual]`

Destructor

### 7.12.3 Member Function Documentation

7.12.3.1 `virtual void glldb::DBConnImp::query ( std::string sql_query ) [pure virtual]`

Runs an SQL query.

Parameters

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

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

### 7.12.3.2 virtual Table glldb::DBConnImp::select ( 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:

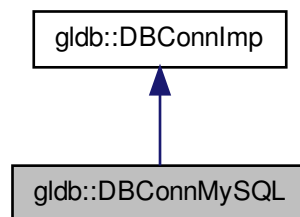
- lib/database/[dbconnimp.h](#)

## 7.13 glldb::DBConnMySQL Class Reference

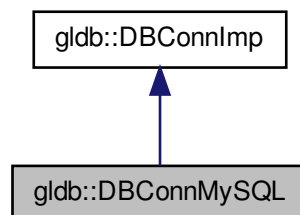
MySQL database implementation class.

```
#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](#) &)
- virtual [~DBConnMySQL](#) ()
- [DBConnMySQL](#) & [operator=](#) (const [DBConnMySQL](#) &)
- virtual void [query](#) (std::string sql\_query)  
*Runs an SQL query.*
- virtual [Table select](#) (std::string query)  
*Runs an SQL SELECT query.*

## Private Attributes

- MySQL \* [m\\_conn](#)

### 7.13.1 Detailed Description

MySQL database implementation class.

### 7.13.2 Constructor & Destructor Documentation

- 7.13.2.1 [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

<a href="#">DBConnCouldNotConnect</a>	If could not connect to database.
---------------------------------------	-----------------------------------

- 7.13.2.2 [gldb::DBConnMySQL::DBConnMySQL](#) ( const [DBConnMySQL](#) & )

Deleted copy constructor

- 7.13.2.3 [DBConnMySQL::~~DBConnMySQL](#) ( ) [virtual]

Destructor

### 7.13.3 Member Function Documentation

#### 7.13.3.1 DBConnMySQL& glldb::DBConnMySQL::operator= ( const DBConnMySQL & )

Deleted assignment operator

#### 7.13.3.2 void DBConnMySQL::query ( std::string *sql\_query* ) [virtual]

Runs an SQL query.

##### Parameters

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

##### Exceptions

<a href="#">DBConnCouldNotQuery</a>	If could not successfully execute query.
-------------------------------------	--

Implements [glldb::DBConnImp](#).

#### 7.13.3.3 Table DBConnMySQL::select ( std::string *query* ) [virtual]

Runs an SQL SELECT query.

##### Parameters

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

##### Returns

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

##### Exceptions

<a href="#">DBConnCouldNotQuery</a>	If could not successfully execute query.
-------------------------------------	--

Implements [glldb::DBConnImp](#).

### 7.13.4 Member Data Documentation

#### 7.13.4.1 MYSQL\* glldb::DBConnMySQL::m\_conn [private]

The initialized MySQL handle.

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

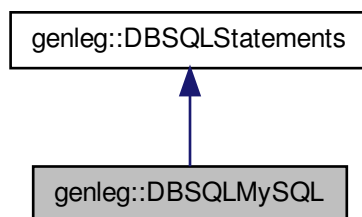
- lib/database\_imp/mysql/[dbconn\\_mysql\\_imp.h](#)
- lib/database\_imp/mysql/[dbconn\\_mysql\\_imp.cpp](#)

## 7.14 genleg::DBSQLMySQL Class Reference

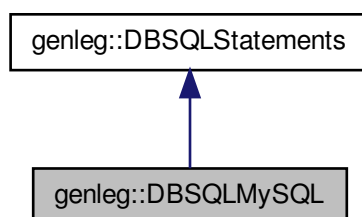
MySQL SQL statements class.

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

Inheritance diagram for genleg::DBSQLMySQL:



Collaboration diagram for genleg::DBSQLMySQL:



## Additional Inherited Members

### 7.14.1 Detailed Description

MySQL SQL statements class.

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

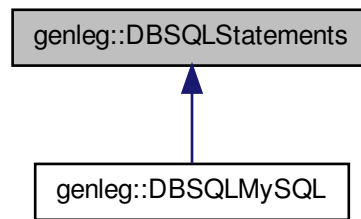
- lib/dbsql/[dbsql\\_mysql.h](#)

## 7.15 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` [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.*

### 7.15.1 Detailed Description

SQL statements class.

### 7.15.2 Constructor & Destructor Documentation

#### 7.15.2.1 `DBSQLStatements::DBSQLStatements ( )`

Constructor

#### 7.15.2.2 `DBSQLStatements::~~DBSQLStatements ( )` [virtual]

Destructor



### 7.15.3 Member Function Documentation

7.15.3.1 `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.

7.15.3.2 `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.

7.15.3.3 `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.

7.15.3.4 `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.

7.15.3.5 `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.

7.15.3.6 `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:

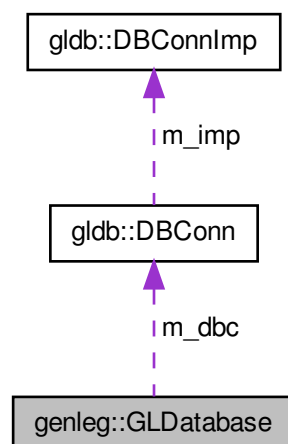
- [lib/dbsql/dbsqlstatements.h](#)
- [lib/dbsql/dbsqlstatements.cpp](#)

## 7.16 genleg::GLDatabase Class Reference

General ledger database class.

```
#include <glatabase.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.*
- [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.*

## Static Public Member Functions

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

## Private Attributes

- [gldb::DBConn](#) [m\\_dbc](#)
- std::shared\_ptr< [DBSQLStatements](#) > [m\\_sql](#)
- const std::vector< std::string > [m\\_tables](#)
- const std::vector< std::string > [m\\_views](#)

### 7.16.1 Detailed Description

General ledger database class.

### 7.16.2 Constructor & Destructor Documentation

- 7.16.2.1 [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

<a href="#">GLDBException</a>	on error.
-------------------------------	-----------

### 7.16.2.2 GLDatabase::~~GLDatabase ( )

Destructor

## 7.16.3 Member Function Documentation

### 7.16.3.1 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.

### 7.16.3.2 void GLDatabase::create\_structure ( )

Creates the database structure.

Exceptions

<a href="#"><i>GLDBException</i></a>	on error.
--------------------------------------	-----------

### 7.16.3.3 void GLDatabase::destroy\_structure ( )

Destroys the database structure.

Exceptions

<a href="#"><i>GLDBException</i></a>	on error.
--------------------------------------	-----------

### 7.16.3.4 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

<a href="#"><i>GLDBException</i></a>	if the user cannot be found.
--------------------------------------	------------------------------

### 7.16.3.5 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

<a href="#"><i>GLDBException</i></a>	if the user cannot be found.
--------------------------------------	------------------------------

## 7.16.3.6 void GLDatabase::load\_sample\_data ( const std::string &amp; 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

<a href="#"><i>GLDBException</i></a>	on error.
--------------------------------------	-----------

## 7.16.4 Member Data Documentation

## 7.16.4.1 glldb::DBConn genleg::GLDatabase::m\_dbc [private]

Database connection

## 7.16.4.2 std::shared\_ptr&lt;DBSQLStatements&gt; genleg::GLDatabase::m\_sql [private]

SQL statements object

## 7.16.4.3 const std::vector&lt;std::string&gt; genleg::GLDatabase::m\_tables [private]

Vector containing database table names

## 7.16.4.4 const std::vector&lt;std::string&gt; 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](#)

## 7.17 genleg::GLDBException Class Reference

Base general ledger database exceptionc class.

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

## Public Member Functions

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

### 7.17.1 Detailed Description

Base general ledger database exceptionc class.

### 7.17.2 Constructor & Destructor Documentation

7.17.2.1 `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:

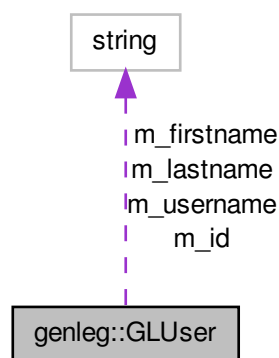
- [lib/gldb/glexception.h](#)

## 7.18 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 [last-name](#), const bool [enabled](#))

Constructor.

- `~GLUser ()`
- `std::string id () const`  
Returns the user ID.
- `std::string username () const`  
Returns the username.
- `std::string firstname () const`  
Returns the user's first name.
- `std::string lastname () const`  
Returns the user's last name.
- `bool enabled () const`  
Returns the user's enabled status.

## Private Attributes

- `std::string m_id`
- `std::string m_username`
- `std::string m_firstname`
- `std::string m_lastname`
- `bool m_enabled`

### 7.18.1 Detailed Description

General ledger user class.

### 7.18.2 Constructor & Destructor Documentation

**7.18.2.1** `GLUser::GLUser ( const std::string id, const std::string username, const std::string firstname, const std::string lastname, 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>enabled</i>	true if user is enabled, false otherwise.

**7.18.2.2** `GLUser::~~GLUser ( )`

Destructor

### 7.18.3 Member Function Documentation

**7.18.3.1** `bool GLUser::enabled ( ) const`

Returns the user's enabled status.

**Returns**

The user's enabled status.

**7.18.3.2   `std::string GLUser::firstname ( ) const`**

Returns the user's first name.

**Returns**

The user's first name.

**7.18.3.3   `std::string GLUser::id ( ) const`**

Returns the user ID.

**Returns**

The user ID.

**7.18.3.4   `std::string GLUser::lastname ( ) const`**

Returns the user's last name.

**Returns**

The user's last name.

**7.18.3.5   `std::string GLUser::username ( ) const`**

Returns the username.

**Returns**

The username.

**7.18.4   Member Data Documentation****7.18.4.1   `bool genleg::GLUser::m_enabled   [private]`**

User's enabled status

**7.18.4.2   `std::string genleg::GLUser::m_firstname   [private]`**

User's first name

**7.18.4.3   `std::string genleg::GLUser::m_id   [private]`**

User ID

**7.18.4.4   `std::string genleg::GLUser::m_lastname   [private]`**

User's last name



## 7.18.4.5 std::string genleg::GLUser::m\_username [private]

Username

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

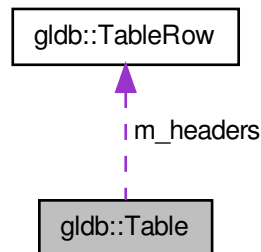
- lib/gldb/gluser.h
- lib/gldb/gluser.cpp

## 7.19 glldb::Table Class Reference

Database table class.

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

Collaboration diagram for glldb::Table:



### Public Member Functions

- [Table](#) (const [TableRow](#) &headers)  
*Constructor.*
- [~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.*
- void [set\\_quoted](#) (std::vector< bool > &vec)  
*Sets the quote flags for the records.*
- 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.*
- 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](#)

### 7.19.1 Detailed Description

Database table class.

### 7.19.2 Constructor & Destructor Documentation

#### 7.19.2.1 [Table::Table](#) ( const [TableRow](#) & *headers* ) `[explicit]`

Constructor.

##### Parameters

<i>headers</i>	<a href="#">Table</a> row containing field names.
----------------	---

#### 7.19.2.2 [Table::~~Table](#) ( )

Destructor

### 7.19.3 Member Function Documentation

#### 7.19.3.1 void [Table::append\\_record](#) ( const [TableRow](#) & *new\_record* )

Appends a record to the table.

##### Parameters

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

#### 7.19.3.2 [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

<a href="#"><i>TableBadInputFile</i></a>	on badly formed input file.
<a href="#"><i>TableCouldNotOpenInputFile</i></a>	on bad filename.

7.19.3.3 `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

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

7.19.3.4 `const TableRow & Table::get_headers ( ) const`

Returns the field names.

## Returns

The field names.

7.19.3.5 `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.

7.19.3.6 `size_t Table::num_fields ( ) const`

Returns the number of fields in each row.

## Returns

The number of fields in each row.

#### 7.19.3.7 `size_t Table::num_records ( ) const`

Returns the number of record in the table.

##### Returns

The number of records in the table.

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

Overloaded index operator.

##### Parameters

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

##### Returns

The selected record.

#### 7.19.3.9 `void Table::set_quoted ( std::vector< bool > & vec )`

Sets the quote flags for the records.

##### Parameters

<code>vec</code>	A vector of bools. The size must match the size of the records.
------------------	---

### 7.19.4 Member Data Documentation

#### 7.19.4.1 `TableRow glldb::Table::m_headers` `[private]`

The names of the fields

#### 7.19.4.2 `std::vector<bool> glldb::Table::m_quoted` `[private]`

A vector to show if fields should be quoted for INSERT

#### 7.19.4.3 `std::vector<TableRow> glldb::Table::m_records` `[private]`

A vector of the records

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

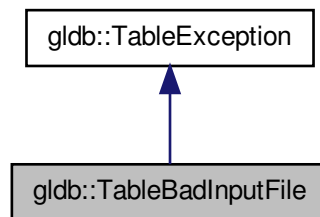
- [lib/database/table.h](#)
- [lib/database/table.cpp](#)

## 7.20 `glldb::TableBadInputFile` Class Reference

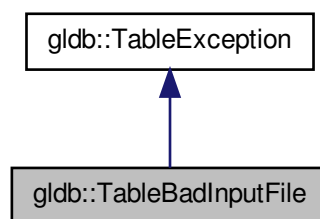
Could not connect to database exception class.

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

Inheritance diagram for glldb::TableBadInputFile:



Collaboration diagram for glldb::TableBadInputFile:



## Public Member Functions

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

### 7.20.1 Detailed Description

Could not connect to database exception class.

### 7.20.2 Constructor & Destructor Documentation

7.20.2.1 `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:

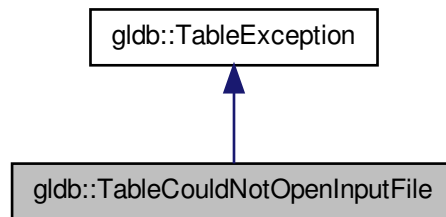
- [lib/database/table.h](#)

## 7.21 glldb::TableCouldNotOpenInputFile Class Reference

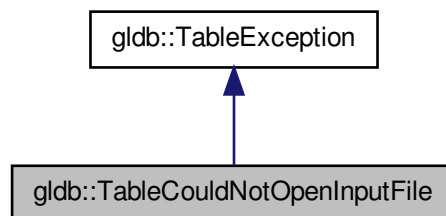
Could not connect to database exception class.

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

Inheritance diagram for glldb::TableCouldNotOpenInputFile:



Collaboration diagram for glldb::TableCouldNotOpenInputFile:



### Public Member Functions

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

#### 7.21.1 Detailed Description

Could not connect to database exception class.

#### 7.21.2 Constructor & Destructor Documentation

7.21.2.1 `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:

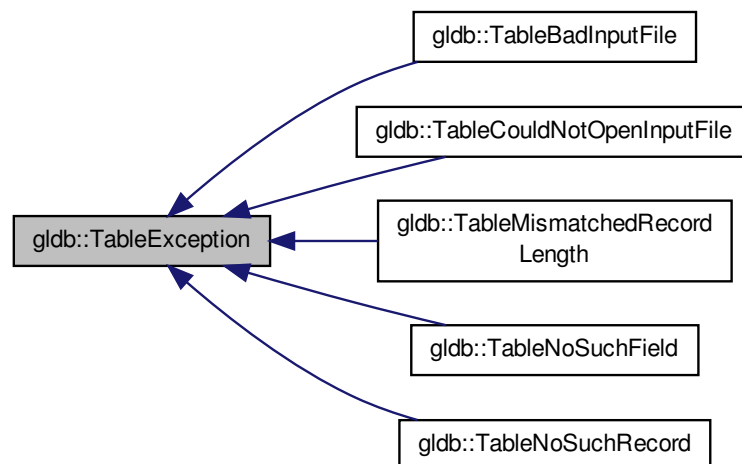
- [lib/database/table.h](#)

## 7.22 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.*

### 7.22.1 Detailed Description

Base database connection exception class.

### 7.22.2 Constructor & Destructor Documentation

### 7.22.2.1 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:

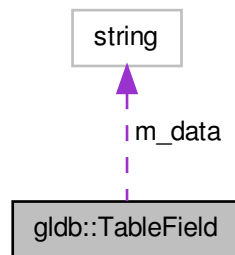
- lib/database/[table.h](#)

## 7.23 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](#) ()
- [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`.*
- 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.*

## 7.23.1 Detailed Description

Database table field class.

## 7.23.2 Constructor & Destructor Documentation

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

Constructor accepting `const char * data`.

#### Parameters

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

### 7.23.2.2 [TableField::TableField](#) ( const std::string & *data* ) [explicit]

Constructor accepting `std::string data`.

#### Parameters

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

### 7.23.2.3 [TableField::~~TableField](#) ( )

Destructor

## 7.23.3 Member Function Documentation

### 7.23.3.1 `size_t` [TableField::length](#) ( ) const

Returns the length of the field.

#### Returns

The length of the field.

### 7.23.3.2 `TableField::operator std::string ( ) const`

Overridden conversion operator.

Returns the field contents as a string.

### 7.23.3.3 `TableField & TableField::operator+= ( const char & c )`

Overridden compound assignment operator.

#### Parameters

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

#### Returns

A reference to the same field.

### 7.23.3.4 `TableField & TableField::operator+= ( const std::string & data )`

Overridden compound assignment operator.

#### Parameters

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

#### Returns

A reference to the same field.

### 7.23.3.5 `TableField & TableField::operator= ( const char * data )`

Overridden assignment operator for `const char *`.

#### Parameters

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

#### Returns

A reference to the same field.

### 7.23.3.6 `TableField & TableField::operator= ( const std::string & data )`

Overridden assignment operator for `std::string`.

#### Parameters

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

#### Returns

A reference to the same field.

7.23.3.7 `char & TableField::operator[] ( const size_t idx )`

Overridden index operator.

## Parameters

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

## Returns

A reference to the character at the specified index.

7.23.3.8 `const char & TableField::operator[] ( const size_t idx ) const`

Overridden index operator.

## Parameters

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

## Returns

A const reference to the character at the specified index.

## 7.23.4 Friends And Related Function Documentation

7.23.4.1 `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*.

## 7.23.5 Member Data Documentation

7.23.5.1 `std::string glldb::TableField::m_data` [*private*]

The field contents

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

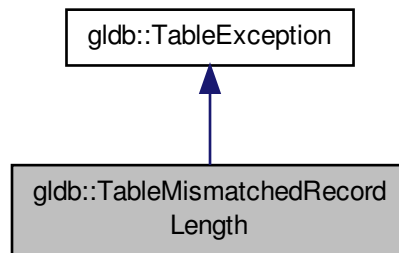
- lib/database/[tablefield.h](#)
- lib/database/[tablefield.cpp](#)

## 7.24 glldb::TableMismatchedRecordLength Class Reference

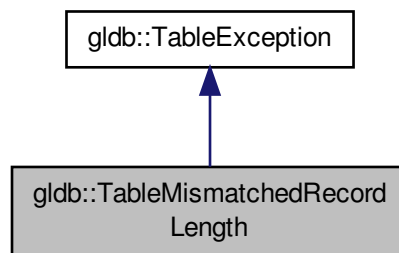
Mismatched record length exception class.

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

Inheritance diagram for `gldb::TableMismatchedRecordLength`:



Collaboration diagram for `gldb::TableMismatchedRecordLength`:



## Public Member Functions

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

*Constructor.*

### 7.24.1 Detailed Description

Mismatched record length exception class.

### 7.24.2 Constructor & Destructor Documentation

- 7.24.2.1 `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:

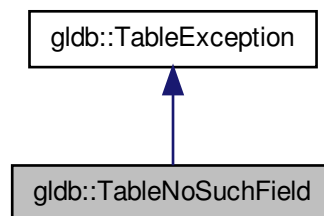
- lib/database/[table.h](#)

## 7.25 glldb::TableNoSuchField Class Reference

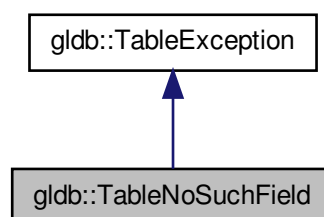
No such field exception class.

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

Inheritance diagram for glldb::TableNoSuchField:



Collaboration diagram for glldb::TableNoSuchField:



### Public Member Functions

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

#### 7.25.1 Detailed Description

No such field exception class.

## 7.25.2 Constructor & Destructor Documentation

7.25.2.1 `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:

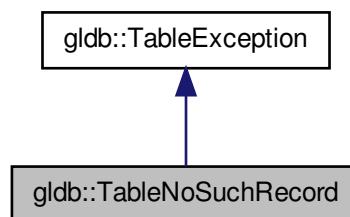
- [lib/database/table.h](#)

## 7.26 glldb::TableNoSuchRecord Class Reference

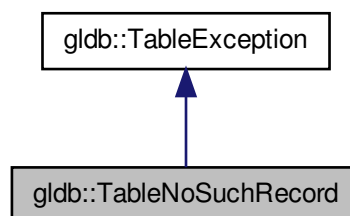
No such record exception class.

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

Inheritance diagram for `gldb::TableNoSuchRecord`:



Collaboration diagram for `gldb::TableNoSuchRecord`:



## Public Member Functions

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

*Constructor.*

### 7.26.1 Detailed Description

No such record exception class.

### 7.26.2 Constructor & Destructor Documentation

#### 7.26.2.1 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](#)

## 7.27 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](#) (std::vector< std::string > &vec)  
*Constructor with string vector.*
- [~TableRow](#) ()
- size\_t [size](#) () const  
*Returns the number of fields.*
- [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](#) (const [TableField](#) &new\_field)  
*Appends a field to the row.*
- void [print](#) (std::ostream &stream) const  
*Prints a row.*
- std::string [record\\_string](#) (const std::vector< bool > &quoted)  
*Creates a comma separated string of fields.*
- std::string [record\\_string](#) ()  
*Creates an unquoted comma separated string of fields.*

## Private Attributes

- `std::vector< TableField > m_fields`

### 7.27.1 Detailed Description

Database table row class.

### 7.27.2 Constructor & Destructor Documentation

#### 7.27.2.1 `TableRow::TableRow ( )`

Default constructor

#### 7.27.2.2 `TableRow::TableRow ( const size_t size ) [explicit]`

Constructor with initial number of fields.

##### Parameters

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

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

Constructor with string vector.

##### Parameters

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

#### 7.27.2.4 `TableRow::~~TableRow ( )`

Destructor

### 7.27.3 Member Function Documentation

#### 7.27.3.1 `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.
------------------	--------------------------------

#### 7.27.3.2 `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.
------------------	--------------------------------



**7.27.3.3 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.
------------------	-----------------------------

**7.27.3.4 TableField & TableRow::operator[] ( const size\_t *idx* )**

Overridden index operator.

**Parameters**

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

**Returns**

A reference to the field at the specified index.

**7.27.3.5 const TableField & TableRow::operator[] ( const size\_t *idx* ) const**

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.

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

Prints a row.

**Parameters**

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

**7.27.3.7 std::string TableRow::record\_string ( const std::vector< bool > & *quoted* )**

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.

#### 7.27.3.8 `std::string TableRow::record_string ( )`

Creates an unquoted comma separated string of fields.

##### Returns

The unquoted comma separated string.

#### 7.27.3.9 `size_t TableRow::size ( ) const`

Returns the number of fields.

##### Returns

The number of fields.

### 7.27.4 Member Data Documentation

#### 7.27.4.1 `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](#)

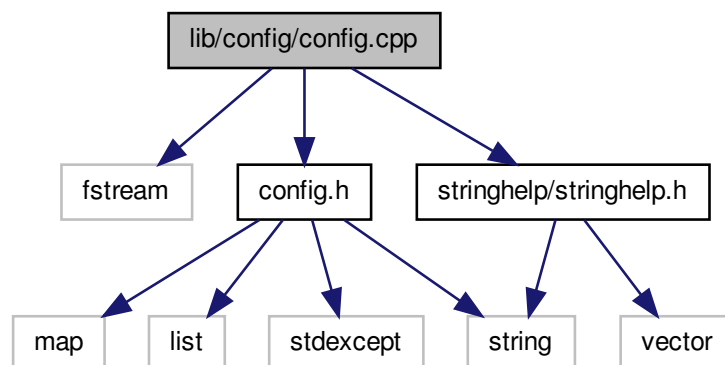
## Chapter 8

# File Documentation

### 8.1 lib/config/config.cpp File Reference

Implementation of program configurations class.

```
#include <fstream>
#include "config.h"
#include "stringhelp/stringhelp.h"
Include dependency graph for config.cpp:
```



#### 8.1.1 Detailed Description

Implementation of program configurations class.

##### Author

Paul Griffiths

##### Copyright

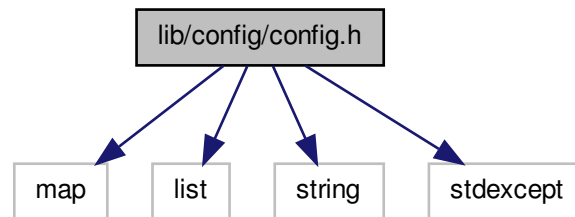
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.2 lib/config/config.h File Reference

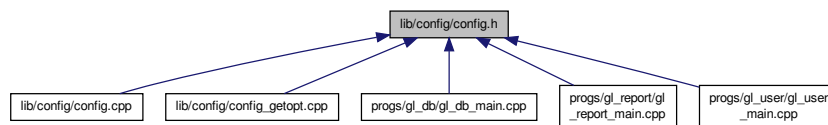
Interface to program configurations class.

```
#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.*

### 8.2.1 Detailed Description

Interface to program configurations class.

#### Author

Paul Griffiths

#### Copyright

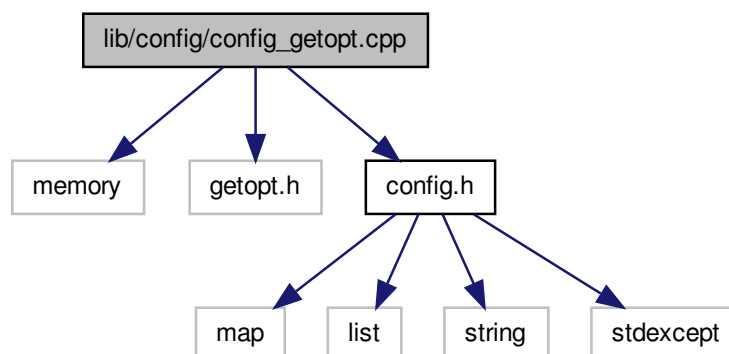
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.3 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`

### 8.3.1 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/>

## 8.3.2 Macro Definition Documentation

### 8.3.2.1 `#define _XOPEN_SOURCE 600`

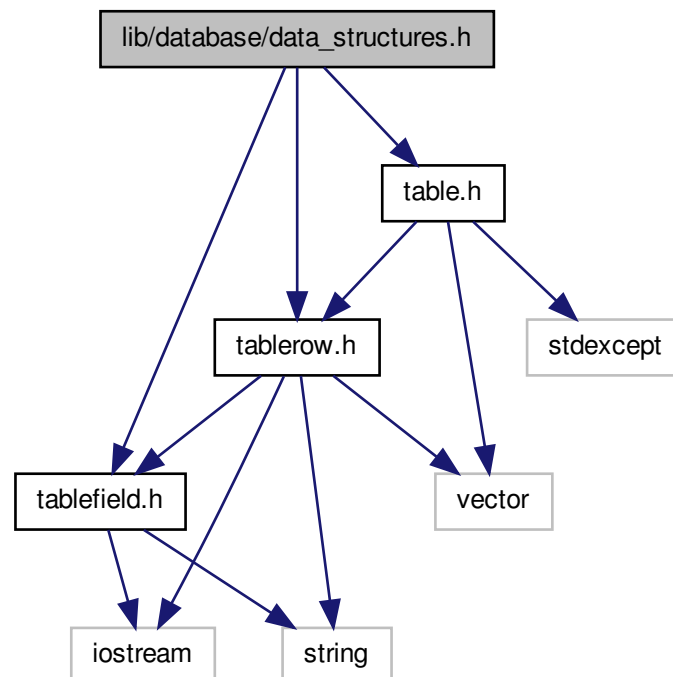
UNIX feature test macro for getopt library

## 8.4 `lib/database/data_structures.h` File Reference

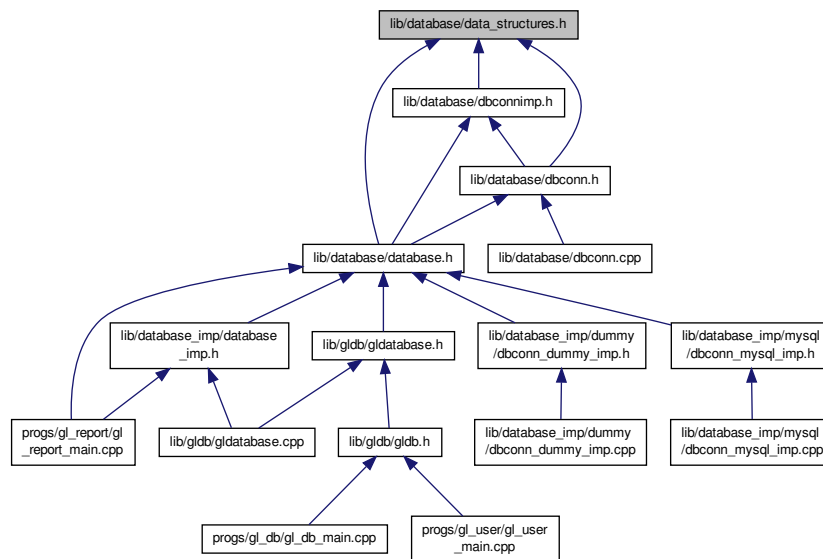
Main interface to database data structures.

```
#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:



### 8.4.1 Detailed Description

Main interface to database data structures.

#### Author

Paul Griffiths

#### Copyright

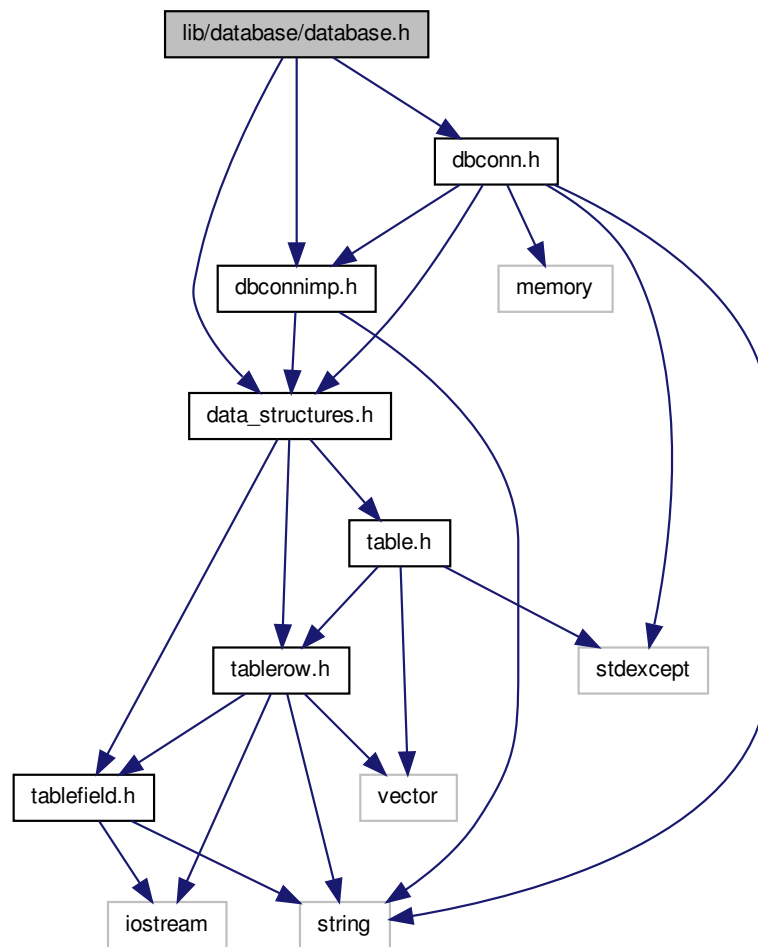
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.5 lib/database/database.h File Reference

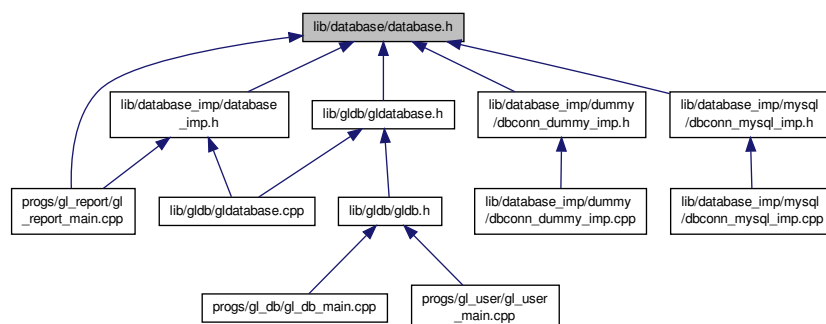
User interface to database functionality.

```
#include "data_structures.h"
#include "dbconnimp.h"
#include "dbconn.h"
```

Include dependency graph for database.h:



This graph shows which files directly or indirectly include this file:





### 8.5.1 Detailed Description

User interface to database functionality.

Author

Paul Griffiths

Copyright

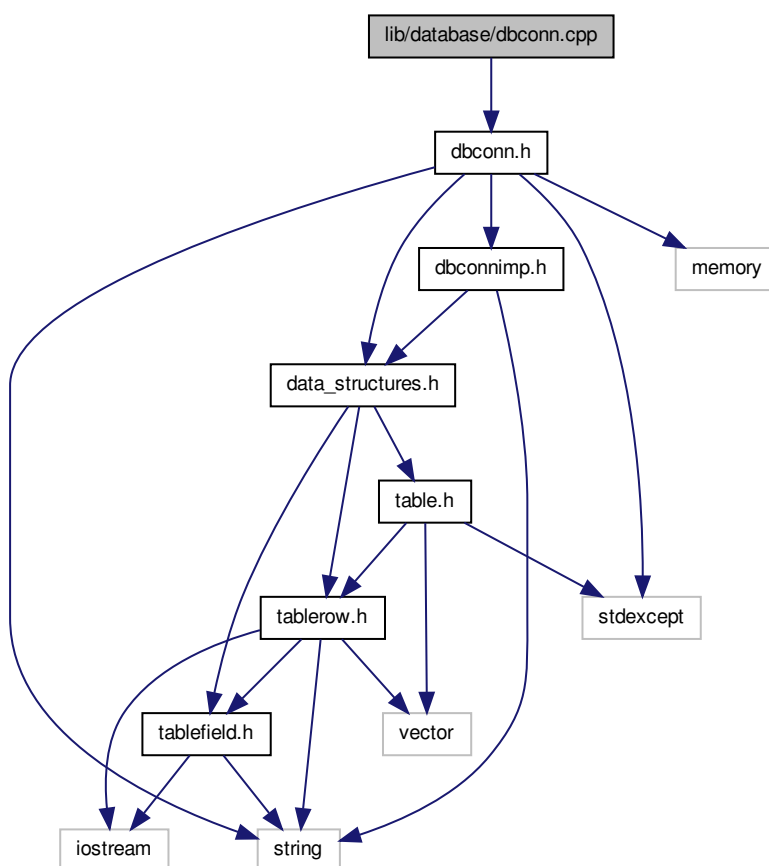
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.6 lib/database/dbconn.cpp File Reference

Implementation of database connection class.

```
#include "dbconn.h"
```

Include dependency graph for dbconn.cpp:



### 8.6.1 Detailed Description

Implementation of database connection class.

## Author

Paul Griffiths

## Copyright

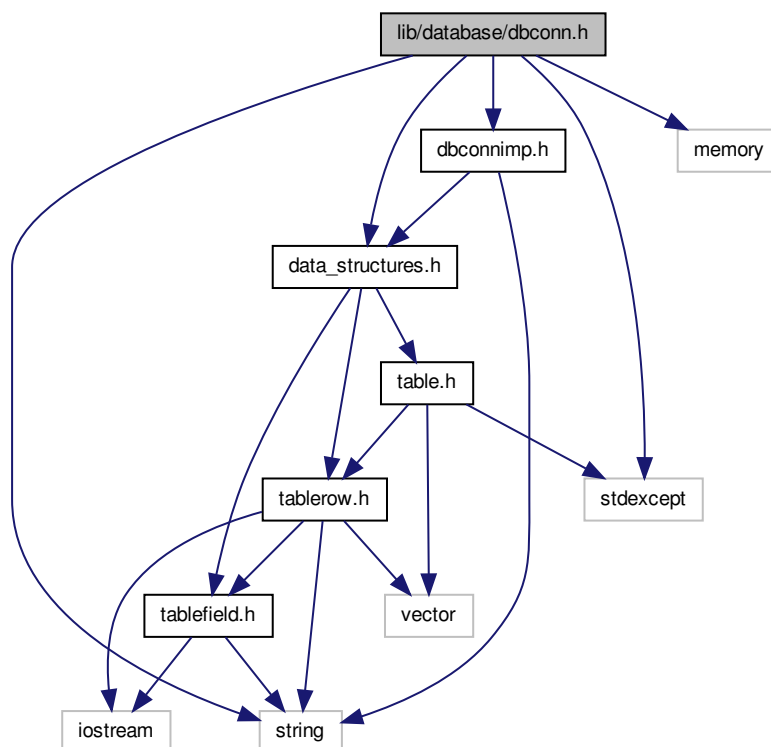
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.7 lib/database/dbconn.h File Reference

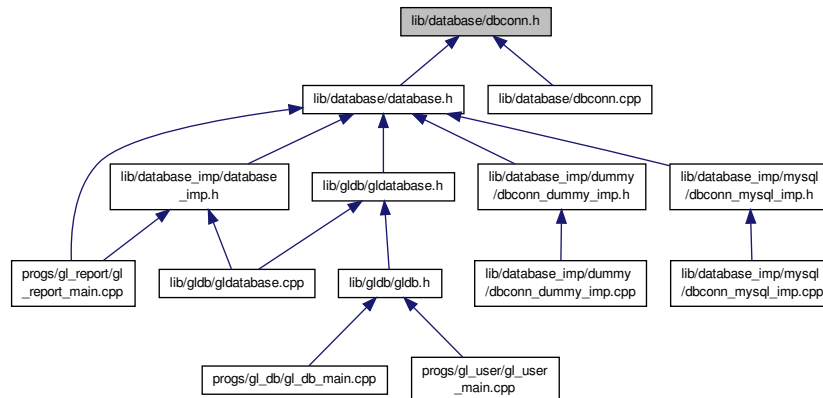
Interface to database connection base class.

```
#include <string>
#include <memory>
#include <stdexcept>
#include "data_structures.h"
#include "dbconnimp.h"
```

Include dependency graph for dbconn.h:



This graph shows which files directly or indirectly include this file:



## 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.*

### 8.7.1 Detailed Description

Interface to database connection base class.

#### Author

Paul Griffiths

#### Copyright

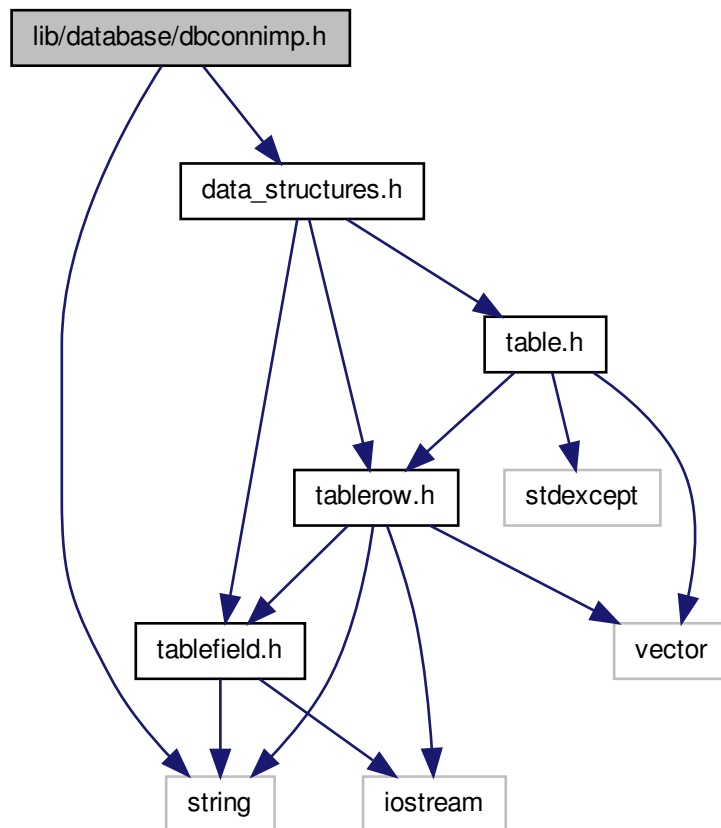
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.8 lib/database/dbconnimp.h File Reference

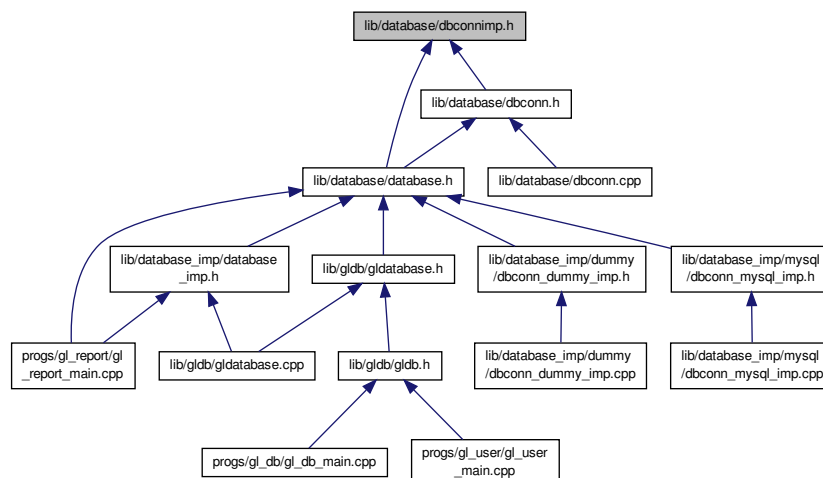
Interface to abstract database implementation base class.

```
#include <string>
#include "data_structures.h"
```

Include dependency graph for dbconnimp.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gldb::DBConnImp](#)  
*Abstract database implementation base class.*

### 8.8.1 Detailed Description

Interface to abstract database implementation base class.

#### Author

Paul Griffiths

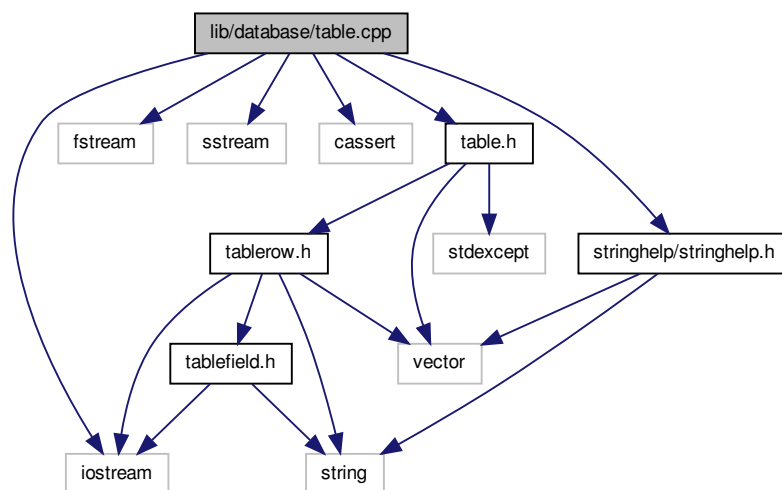
#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.9 lib/database/table.cpp File Reference

Implementation of database table data structure.

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <cassert>
#include "table.h"
#include "stringhelp/stringhelp.h"
Include dependency graph for table.cpp:
```



### 8.9.1 Detailed Description

Implementation of database table data structure.

## Author

Paul Griffiths

## Copyright

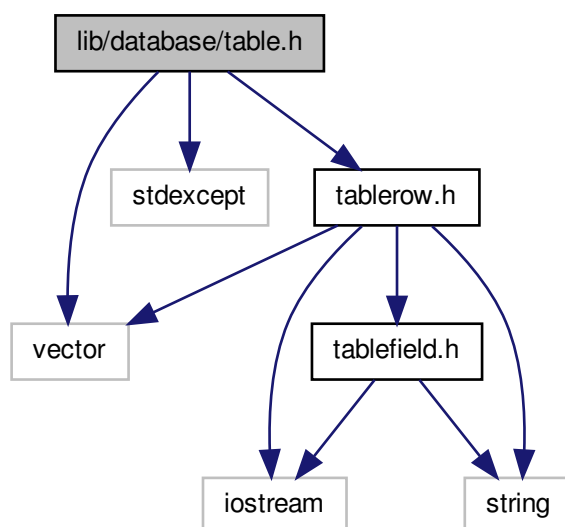
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.10 lib/database/table.h File Reference

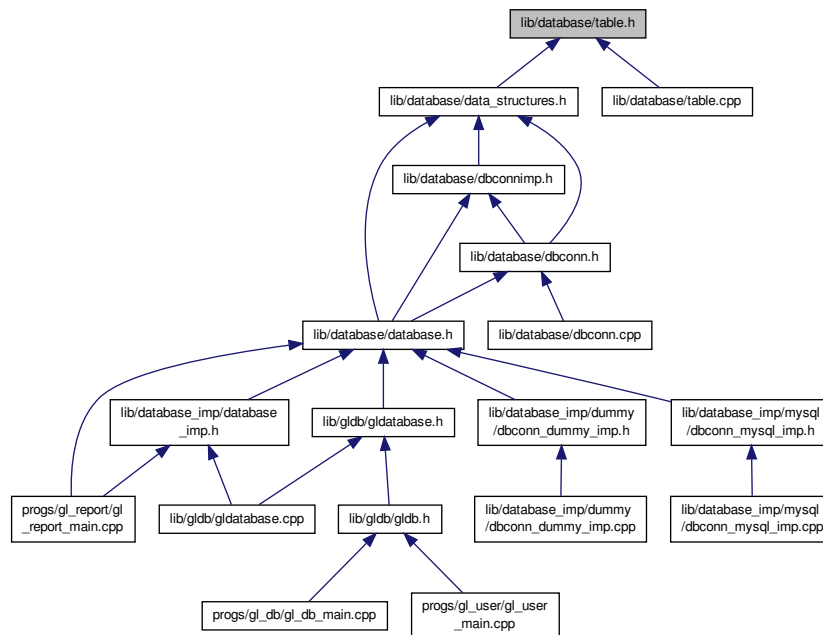
Interface to database table data structure.

```
#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`  
*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.*

### 8.10.1 Detailed Description

Interface to database table data structure.

#### Author

Paul Griffiths

#### Copyright

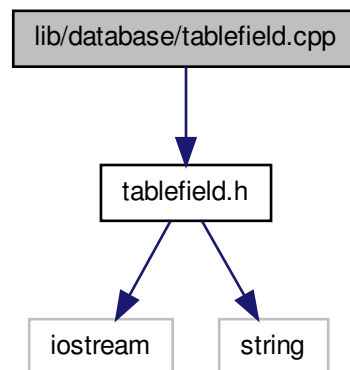
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.11 lib/database/tablefield.cpp File Reference

Implementation of database table field class.

```
#include "tablefield.h"
```

Include dependency graph for tablefield.cpp:



### 8.11.1 Detailed Description

Implementation of database table field class.

#### Author

Paul Griffiths

#### Copyright

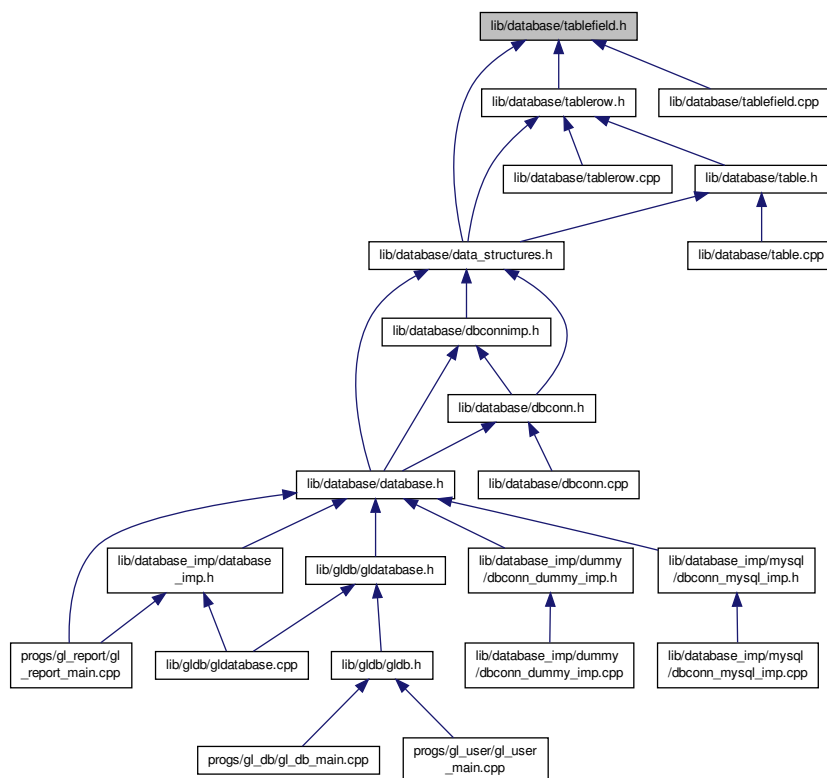
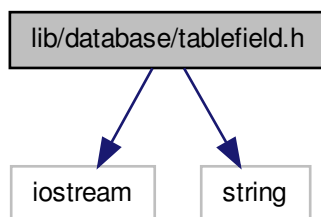
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.12 lib/database/tablefield.h File Reference

Interface to database table field class.

```
#include <iostream>
#include <string>
```





- class `gldb::TableField`  
*Database table field class.*

- `std::ostream & gldb::operator<< (std::ostream &out, const TableField &field)`

Overridden << operator for printing a field.

### 8.12.1 Detailed Description

Interface to database table field class.

#### Author

Paul Griffiths

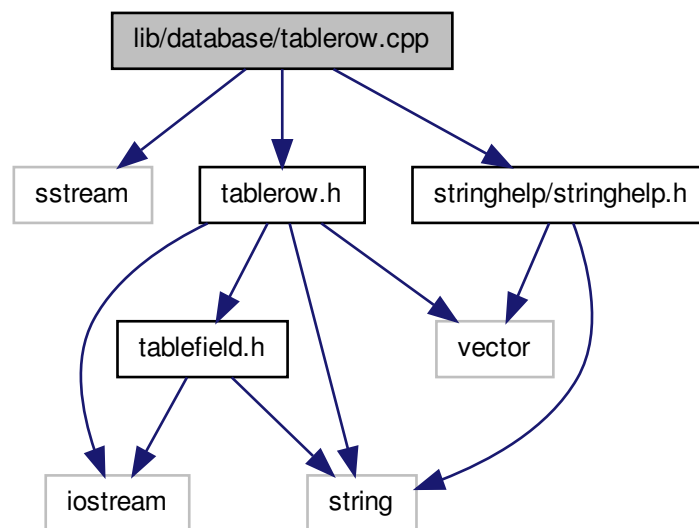
#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.13 lib/database/ablerow.cpp File Reference

Implementation of database table row data structure.

```
#include <sstream>
#include "ablerow.h"
#include "stringhelp/stringhelp.h"
Include dependency graph for ablerow.cpp:
```



### 8.13.1 Detailed Description

Implementation of database table row data structure.

#### Author

Paul Griffiths

### Copyright

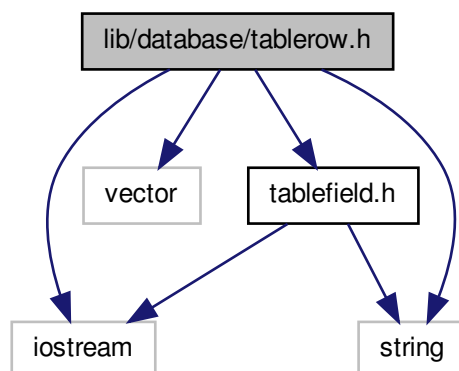
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.14 lib/database/tablerow.h File Reference

Interface to database table row data structure.

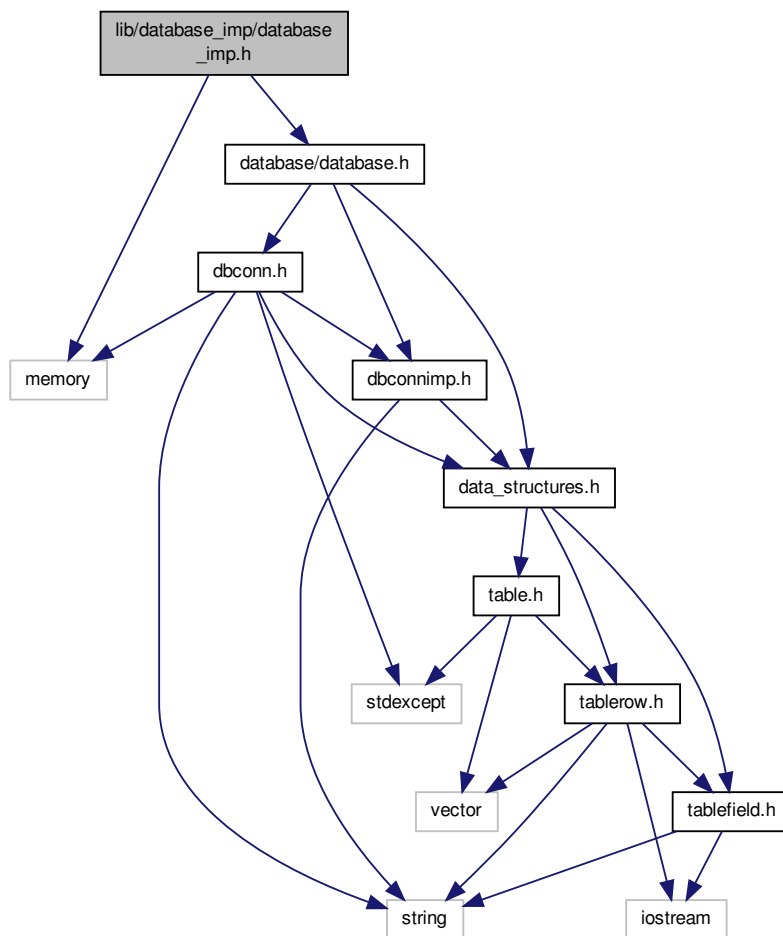
```
#include <iostream>
#include <vector>
#include <string>
#include "tablefield.h"
```

Include dependency graph for tablerow.h:

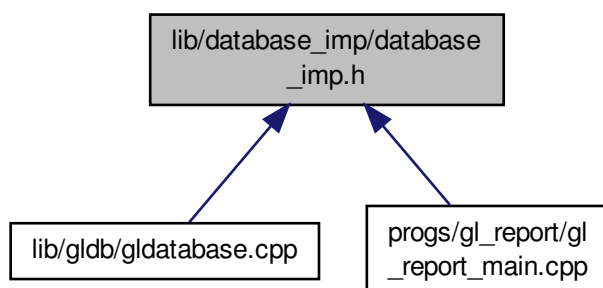




Include dependency graph for database\_imp.h:



This graph shows which files directly or indirectly include this file:



## Functions

- `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.*

- std::string `glldb::get_database_type` ()

*Returns the name of the compiled-in database type.*

### 8.15.1 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/>

## 8.16 lib/database\_imp/dummy/dbconn\_dummy\_imp.cpp File Reference

Implementation of Dummy database connection implementation class.

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

```

graph TD
    Root["lib/database_imp/dummy/dbconn_dummy_imp.cpp"] --> sstream["sstream"]
    Root --> dbconn_dummy_imp_h["dbconn_dummy_imp.h"]
    dbconn_dummy_imp_h --> database_database_h["database/database.h"]
    database_database_h --> dbconn_h["dbconn.h"]
    database_database_h --> memory["memory"]
    database_database_h --> dbconnimp_h["dbconnimp.h"]
    database_database_h --> data_structures_h["data_structures.h"]
    database_database_h --> string["string"]
    dbconn_h --> memory
    dbconn_h --> stdexcept["stdexcept"]
    dbconn_h --> vector["vector"]
    dbconn_h --> table_h["table.h"]
    dbconn_h --> tablerow_h["tablerow.h"]
    dbconn_h --> string
    memory --> string
    dbconnimp_h --> data_structures_h
    dbconnimp_h --> string
    data_structures_h --> table_h
    data_structures_h --> tablerow_h
    data_structures_h --> tablefield_h["tablefield.h"]
    data_structures_h --> string
    table_h --> vector
    table_h --> string
    tablerow_h --> vector
    tablerow_h --> tablefield_h
    tablerow_h --> string
    tablefield_h --> iostream["iostream"]
    tablefield_h --> string
    string --> iostream

```

### Implementation of Dummy database connection implementation class.

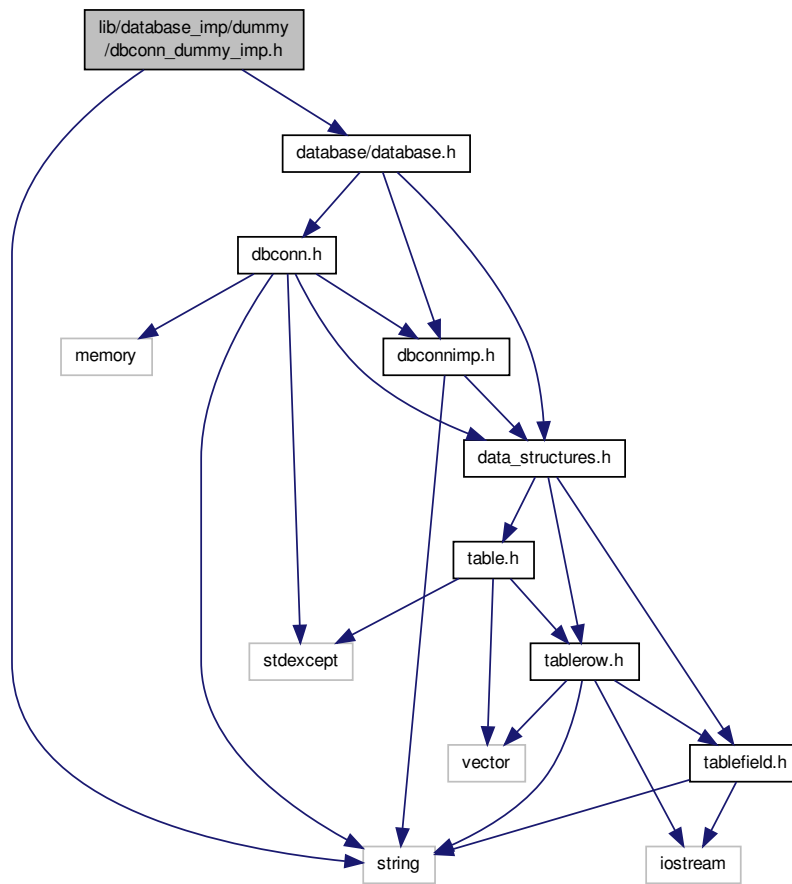
Paul Griffiths

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

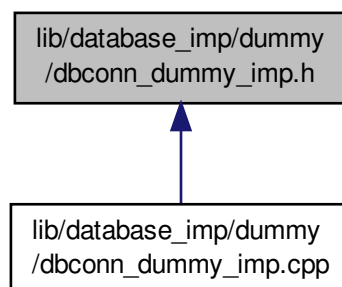
Interface to dummy database connection implementation class.

```
#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.*

### 8.17.1 Detailed Description

Interface to dummy database connection implementation class.

#### Author

Paul Griffiths

#### Copyright

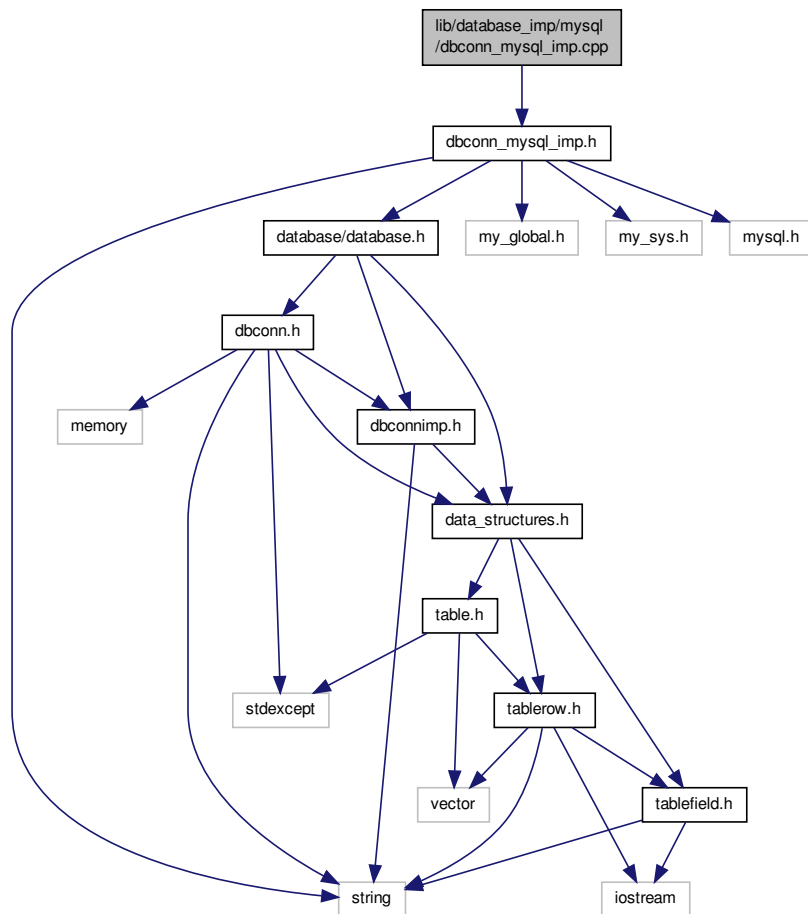
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.18 lib/database\_imp/mysql/dbconn\_mysql\_imp.cpp File Reference

Implementation of MySQL database connection implementation class.

```
#include "dbconn_mysql_imp.h"
```

Include dependency graph for dbconn\_mysql\_imp.cpp:



### 8.18.1 Detailed Description

Implementation of MySQL database connection implementation class.

#### Author

Paul Griffiths

#### Copyright

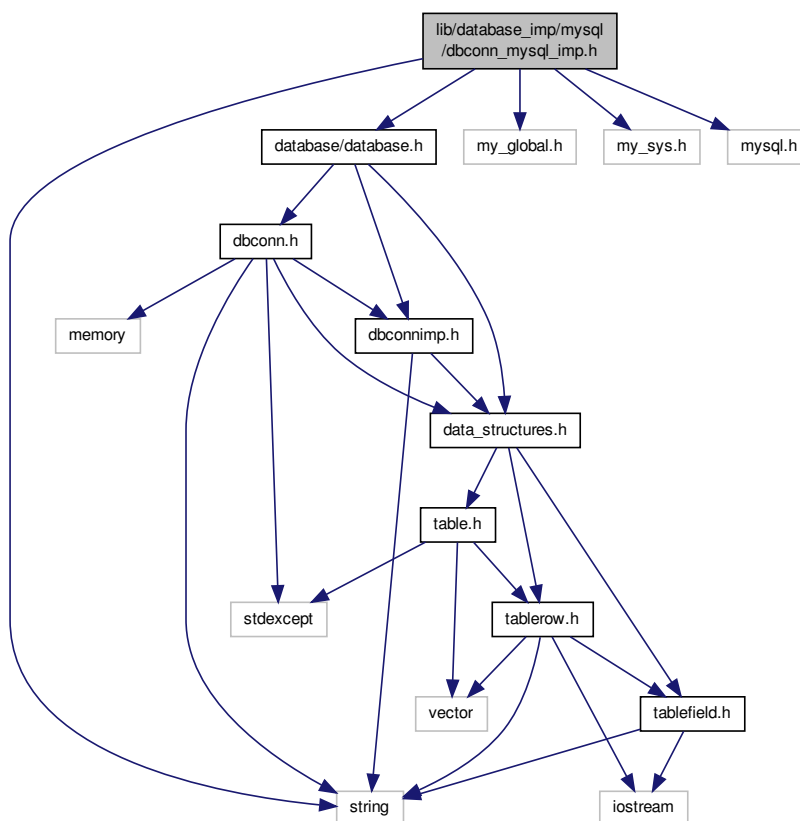
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.19 lib/database\_imp/mysql/dbconn\_mysql\_imp.h File Reference

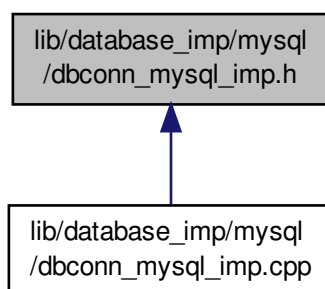
Interface to MySQL database connection implementation class.

```
#include <string>
#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.*

### 8.19.1 Detailed Description

Interface to MySQL database connection implementation class.

#### Author

Paul Griffiths

#### Copyright

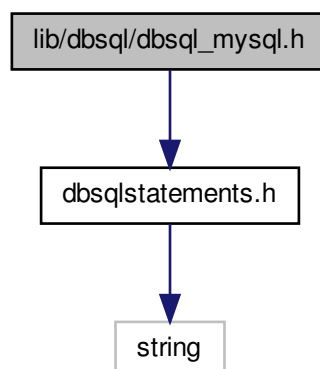
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.20 lib/dbsql/dbsql\_mysql.h File Reference

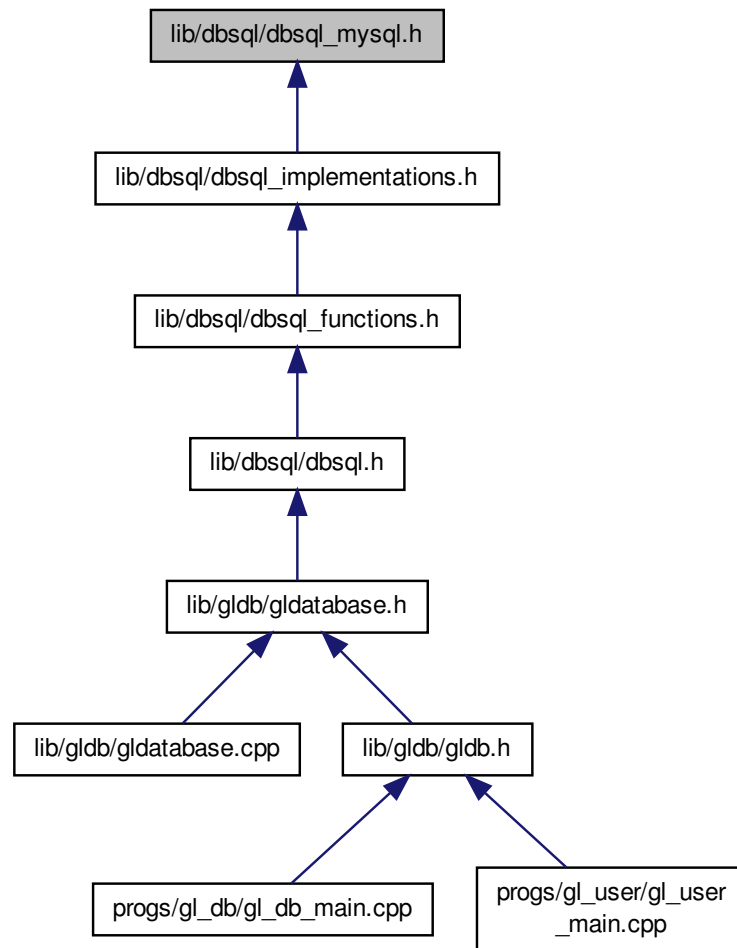
Interface to MySQL SQL statement class.

```
#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.*

### 8.20.1 Detailed Description

Interface to MySQL SQL statement class. 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/>

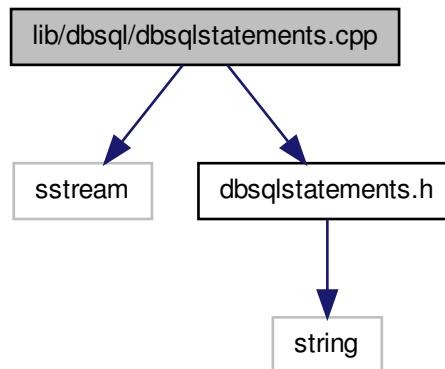
## 8.21 lib/dbsql/dbsqlstatements.cpp File Reference

Implementation of SQL statement class.

```
#include <sstream>
```

```
#include "dbsqlstatements.h"
```

Include dependency graph for dbsqlstatements.cpp:



### 8.21.1 Detailed Description

Implementation of SQL statement class. Implementation of SQL statement class

#### Author

Paul Griffiths

### Copyright

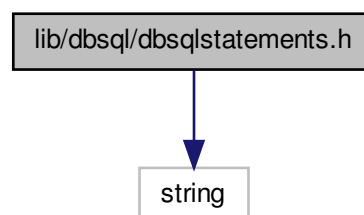
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.22 lib/dbsql/dbsqlstatements.h File Reference

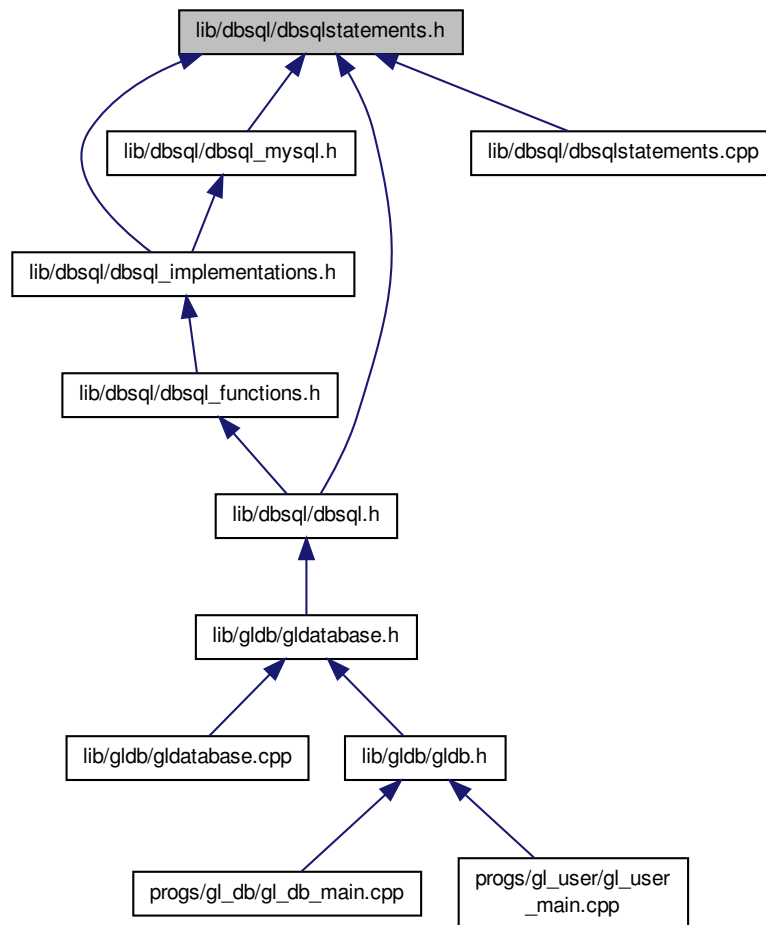
Interface to SQL statement class.

```
#include <string>
```

Include dependency graph for `dbsqlstatements.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [genleg::DBSQLStatements](#)

*SQL statements class.*

### 8.22.1 Detailed Description

Interface to SQL statement class.

#### Author

Paul Griffiths

#### Copyright

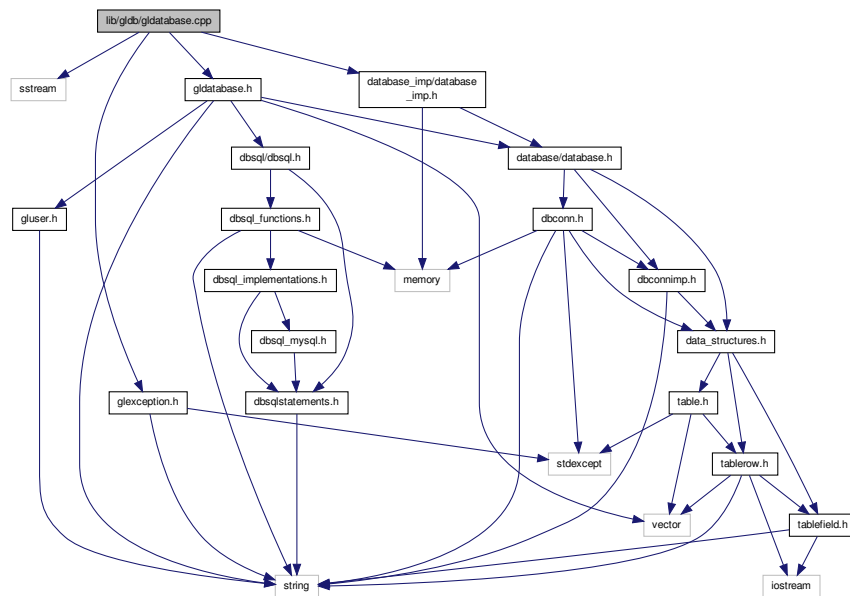
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>



## 8.23 lib/gldb/gldatabase.cpp File Reference

### Implementation of General Ledger database class.

```
#include <sstream>
#include "glatabase.h"
#include "glexception.h"
#include "database_imp/database_imp.h"
Include dependency graph for glatabase.cpp:
```



## Functions

- **m\_views** ({"current\_trial\_balance","check\_total","all\_jes"})

### 8.23.1 Detailed Description

### Implementation of General Ledger database class.

**Author**

Paul Griffiths

**Copyright**

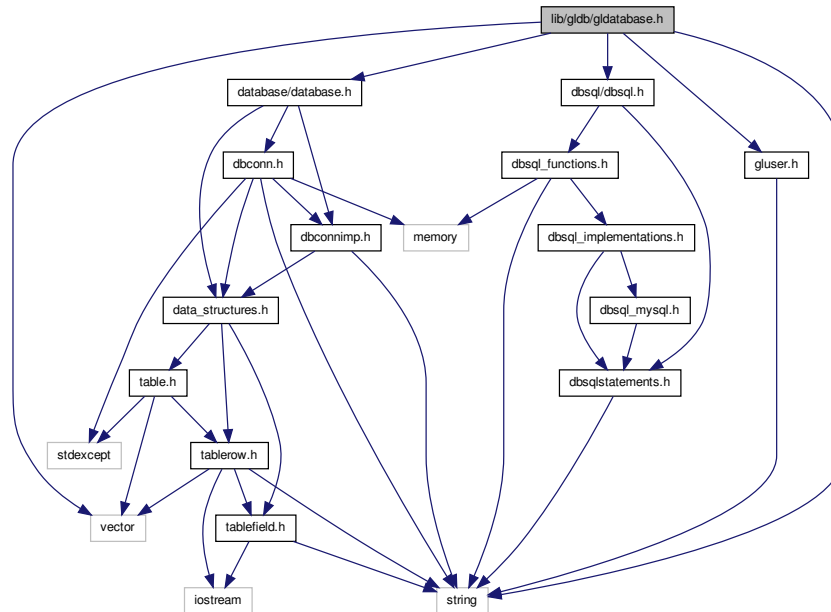
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.24 lib/gldb/gldatabase.h File Reference

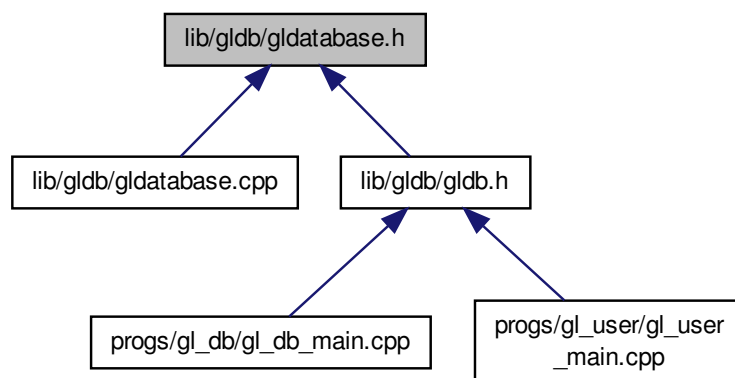
Interface to General Ledger database class.

```
#include <vector>
#include <string>
#include "database/database.h"
#include "dbsql/dbsql.h"
#include "gluser.h"
```

Include dependency graph for glatabase.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [genleg::GLDatabase](#)  
*General ledger database class.*

### 8.24.1 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/>

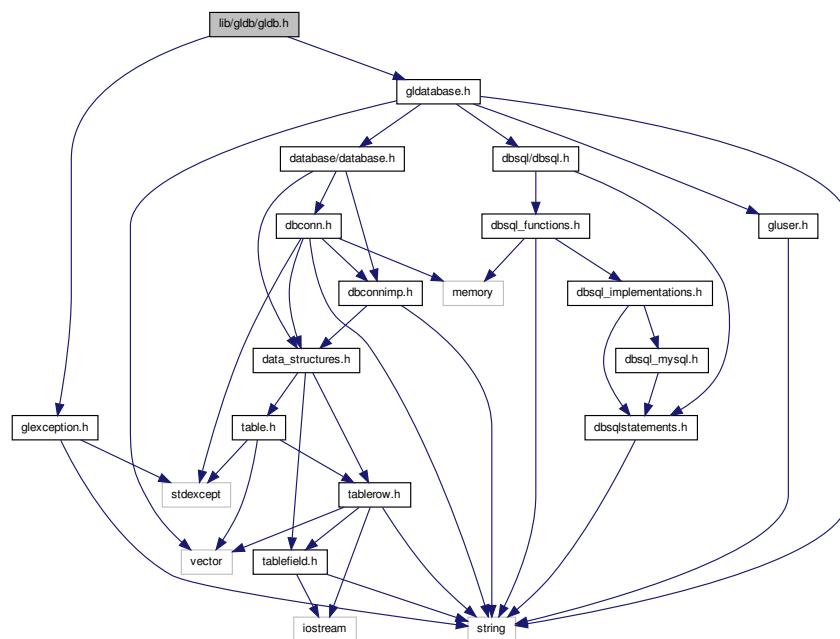
## 8.25 lib/gldb/gldb.h File Reference

User interface to General Ledger database module.

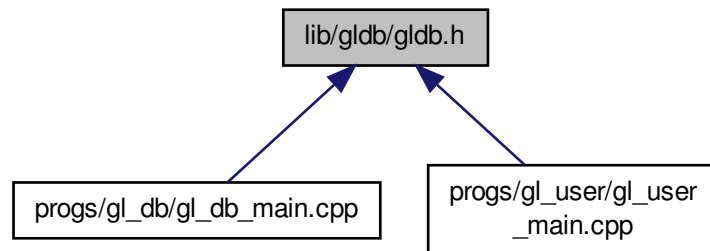
```
#include "glexception.h"
```

```
#include "gldatabase.h"
```

Include dependency graph for gldb.h:



This graph shows which files directly or indirectly include this file:



### 8.25.1 Detailed Description

User interface to General Ledger database module.

#### Author

Paul Griffiths

#### Copyright

Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

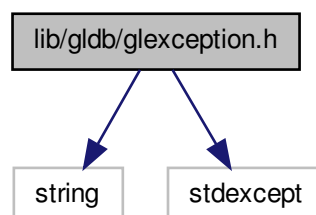
## 8.26 lib/gldb/glexception.h File Reference

Interface to General Ledger base exception class.

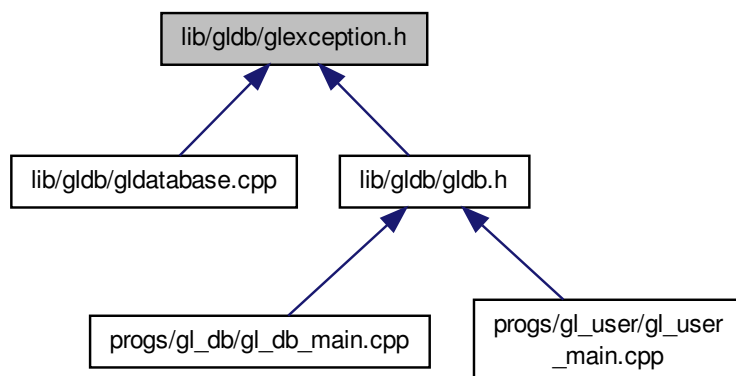
```
#include <string>
```

```
#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.*

### 8.26.1 Detailed Description

Interface to General Ledger base exception class.

#### Author

Paul Griffiths

### Copyright

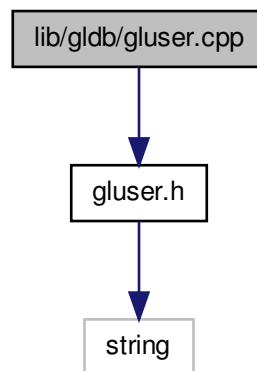
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.27 lib/gdb/gluser.cpp File Reference

Implementation of user class.

```
#include "gluser.h"
```

Include dependency graph for gluser.cpp:



### 8.27.1 Detailed Description

Implementation of user class. Implementation of user class

### Author

Paul Griffiths

### Copyright

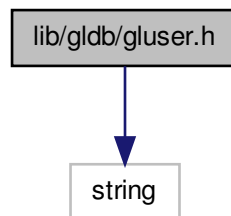
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.28 lib/gdb/gluser.h File Reference

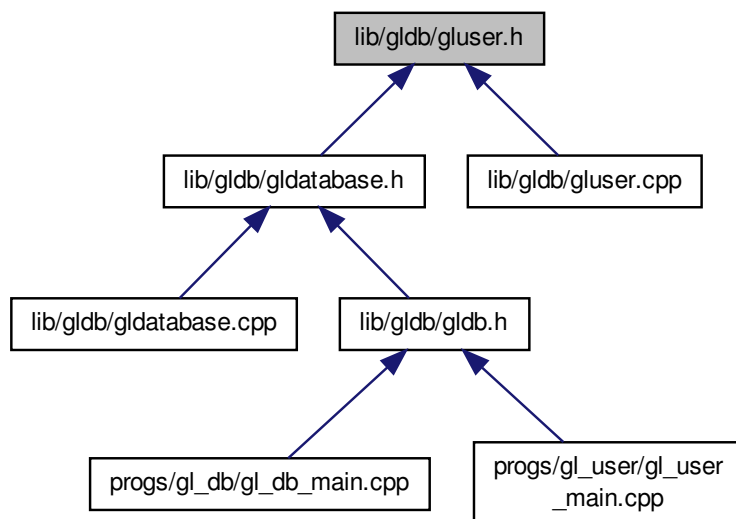
Interface to user class.

```
#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.*

### 8.28.1 Detailed Description

Interface to user class. Interface to user class

**Author**

Paul Griffiths

**Copyright**

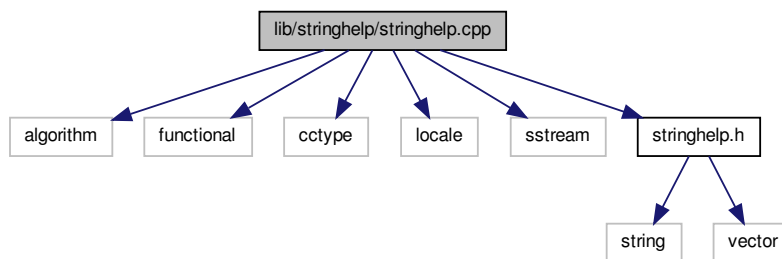
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.29 lib/stringhelp/stringhelp.cpp File Reference

Implementation of string helper functions.

```
#include <algorithm>
#include <functional>
#include <cctype>
#include <locale>
#include <sstream>
#include "stringhelp.h"
```

Include dependency graph for stringhelp.cpp:



### 8.29.1 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/>

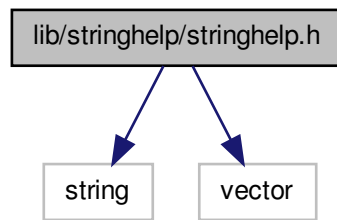
## 8.30 lib/stringhelp/stringhelp.h File Reference

Interface to string helper functions.

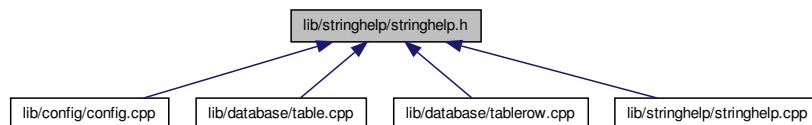
```
#include <string>
#include <vector>
```



Include dependency graph for stringhelp.h:



This graph shows which files directly or indirectly include this file:



## Functions

- `std::string & pgstring::trim_front (std::string &s)`  
*Trims leading whitespace from a string.*
- `std::string & pgstring::trim_back (std::string &s)`  
*Trims trailing whitespace from a string.*
- `std::string & pgstring::trim (std::string &s)`  
*Trims leading and trailing whitespace from a string.*
- `std::vector< std::string > pgstring::split (const std::string &s, const char delim)`  
*Splits a delimited string into tokens.*
- `std::vector< std::string > & pgstring::split (std::vector< std::string > &vec, const std::string &s, const char delim)`  
*Splits a delimited string into tokens.*
- `bool pgstring::next_content_line (std::istream &if, std::string &s)`  
*Gets the next content line from a stream.*
- `std::vector< std::string > & pgstring::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 > > & pgstring::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 & pgstring::join (std::vector< std::string > &vec, std::string &s, const char delim)`  
*Joins a vector of strings into a delimited line.*
- `bool pgstring::replace (std::string &str, const std::string &from, const std::string &to)`  
*Replaces a substring with another string.*



*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\_db"

*Static variable for program name.*

### 8.31.1 Detailed Description

Main functionality for gl\_db program.

#### Author

Paul Griffiths

#### Copyright

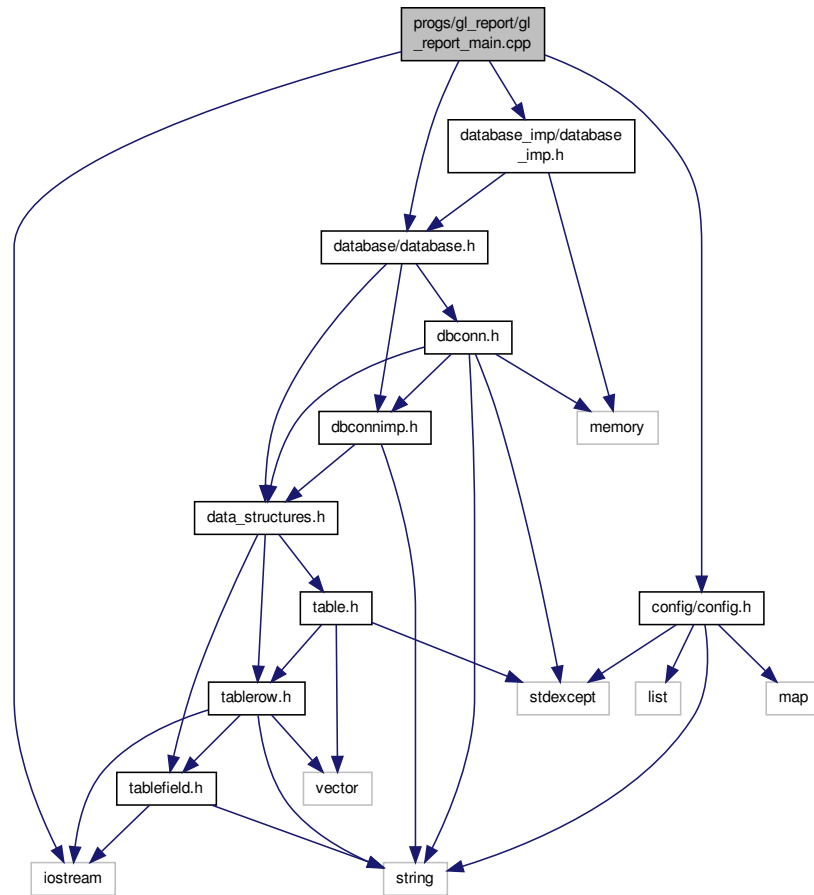
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.32 progs/gl\_report/gl\_report\_main.cpp File Reference

Main functionality for gl\_report program.

```
#include <iostream>
#include "database/database.h"
#include "database_imp/database_imp.h"
#include "config/config.h"
```

Include dependency graph for `gl_report_main.cpp`:



## Functions

- static void `set_configuration` (`genleg::Config` &config, int argc, char \*argv[])  
*Sets program configuration options.*
- 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\_report"  
*Static variable for program name.*

### 8.32.1 Detailed Description

Main functionality for gl\_report program.

Author

Paul Griffiths

Copyright

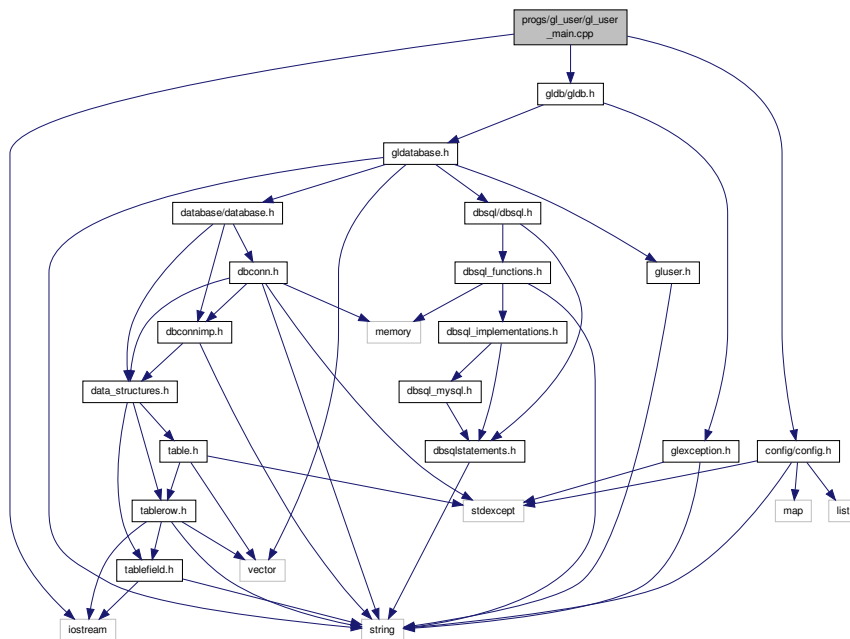
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 8.33 progs/gl\_user/gl\_user\_main.cpp File Reference

Main functionality for gl\_user program.

```
#include <iostream>
#include "gldb/gldb.h"
#include "config/config.h"
```

Include dependency graph for gl\_user\_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 \* `programe` = "gl\_user"  
*Static variable for program name.*

### 8.33.1 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/>

### 8.33.2 Function Documentation

#### 8.33.2.1 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.

#### 8.33.2.2 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.

**8.33.2.3 static std::string login ( void ) [static]**

Gets a password from the terminal.

**Returns**

The password.

**8.33.2.4 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.

**8.33.2.5 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> .

# Index

- ~Config
  - genleg::Config, [23](#)
- ~DBConnDummy
  - gldb::DBConnDummy, [37](#)
- ~DBConnImp
  - gldb::DBConnImp, [39](#)
- ~DBConnMySQL
  - gldb::DBConnMySQL, [41](#)
- ~DBSQLStatements
  - genleg::DBSQLStatements, [44](#)
- ~GLDatabase
  - genleg::GLDatabase, [47](#)
- ~GLUser
  - genleg::GLUser, [51](#)
- ~Table
  - gldb::Table, [54](#)
- ~TableField
  - gldb::TableField, [61](#)
- ~TableRow
  - gldb::TableRow, [68](#)
- \_XOPEN\_SOURCE
  - config\_getopt.cpp, [74](#)
- add\_cmdline\_option
  - genleg::Config, [24](#)
- append\_field
  - gldb::TableRow, [68](#)
- append\_record
  - gldb::Table, [54](#)
- backend
  - genleg::GLDatabase, [48](#)
- check\_db\_parameters
  - Database program., [20](#)
  - gl\_user\_main.cpp, [114](#)
- check\_help\_and\_version
  - Database program., [20](#)
  - gl\_user\_main.cpp, [114](#)
- Config
  - genleg::Config, [23](#)
- config\_getopt.cpp
  - \_XOPEN\_SOURCE, [74](#)
- ConfigBadConfigFile
  - genleg::ConfigBadConfigFile, [26](#)
- ConfigBadOption
  - genleg::ConfigBadOption, [27](#)
- ConfigCouldNotOpenFile
  - genleg::ConfigCouldNotOpenFile, [29](#)
- ConfigException
  - genleg::ConfigException, [30](#)
- ConfigOptionNotSet
  - genleg::ConfigOptionNotSet, [31](#)
- create\_from\_file
  - gldb::Table, [54](#)
- create\_structure
  - genleg::GLDatabase, [48](#)
- create\_table
  - genleg::DBSQLStatements, [45](#)
- create\_view
  - genleg::DBSQLStatements, [45](#)
- DBConn
  - gldb::DBConn, [32](#)
- DBConnCouldNotConnect
  - gldb::DBConnCouldNotConnect, [34](#)
- DBConnCouldNotQuery
  - gldb::DBConnCouldNotQuery, [35](#)
- DBConnDummy
  - gldb::DBConnDummy, [36, 37](#)
- DBConnException
  - gldb::DBConnException, [38](#)
- DBConnImp
  - gldb::DBConnImp, [39](#)
- DBConnMySQL
  - gldb::DBConnMySQL, [41](#)
- DBSQLStatements
  - genleg::DBSQLStatements, [44](#)
- Database interaction module, [12](#)
  - get\_connection, [13](#)
  - get\_database\_type, [13](#)
- Database program., [20](#)
  - check\_db\_parameters, [20](#)
  - check\_help\_and\_version, [20](#)
  - login, [21](#)
  - main, [21](#)
  - set\_configuration, [21](#)
- destroy\_structure
  - genleg::GLDatabase, [48](#)
- drop\_table
  - genleg::DBSQLStatements, [45](#)
- drop\_view
  - genleg::DBSQLStatements, [45](#)
- enabled
  - genleg::GLUser, [51](#)
- firstname
  - genleg::GLUser, [52](#)
- GLDBException



- genleg::GLDBException, 50
- GLDatabase
  - genleg::GLDatabase, 47
- GLUser
  - genleg::GLUser, 51
- General Ledger database module., 11
- General purpose helpers., 16
  - split, 16
  - trim, 16
  - trim\_back, 17
  - trim\_front, 17
- genleg::Config, 23
  - ~Config, 23
  - add\_cmdline\_option, 24
  - Config, 23
  - is\_set, 24
  - m\_opts\_set, 25
  - m\_opts\_supp, 25
  - populate\_from\_cmdline, 24
  - populate\_from\_file, 25
- genleg::ConfigBadConfigFile, 25
  - ConfigBadConfigFile, 26
- genleg::ConfigBadOption, 27
  - ConfigBadOption, 27
- genleg::ConfigCouldNotOpenFile, 28
  - ConfigCouldNotOpenFile, 29
- genleg::ConfigException, 29
  - ConfigException, 30
- genleg::ConfigOptionNotSet, 30
  - ConfigOptionNotSet, 31
- genleg::DBSQLMySQL, 42
- genleg::DBSQLStatements, 43
  - ~DBSQLStatements, 44
  - create\_table, 45
  - create\_view, 45
  - DBSQLStatements, 44
  - drop\_table, 45
  - drop\_view, 45
  - user\_by\_id, 45
  - user\_by\_username, 46
- genleg::GLDBException, 49
  - GLDBException, 50
- genleg::GLDatabase, 46
  - ~GLDatabase, 47
  - backend, 48
  - create\_structure, 48
  - destroy\_structure, 48
  - GLDatabase, 47
  - get\_user\_by\_id, 48
  - get\_user\_by\_username, 48
  - load\_sample\_data, 49
  - m\_dbc, 49
  - m\_sql, 49
  - m\_tables, 49
  - m\_views, 49
- genleg::GLUser, 50
  - ~GLUser, 51
  - enabled, 51
  - firstname, 52
  - GLUser, 51
  - id, 52
  - lastname, 52
  - m\_enabled, 52
  - m\_firstname, 52
  - m\_id, 52
  - m\_lastname, 52
  - m\_username, 52
  - username, 52
- get\_connection
  - Database interaction module, 13
- get\_database\_type
  - Database interaction module, 13
- get\_field
  - gldb::Table, 55
- get\_headers
  - gldb::Table, 55
- get\_user\_by\_id
  - genleg::GLDatabase, 48
- get\_user\_by\_username
  - genleg::GLDatabase, 48
- gl\_user\_main.cpp
  - check\_db\_parameters, 114
  - check\_help\_and\_version, 114
  - login, 115
  - main, 115
  - set\_configuration, 115
- gldb::DBConn, 31
  - DBConn, 32
  - m\_imp, 33
  - operator=, 32
  - query, 32
  - select, 32
- gldb::DBConnCouldNotConnect, 33
  - DBConnCouldNotConnect, 34
- gldb::DBConnCouldNotQuery, 34
  - DBConnCouldNotQuery, 35
- gldb::DBConnDummy, 35
  - ~DBConnDummy, 37
  - DBConnDummy, 36, 37
  - operator=, 37
  - select, 37
- gldb::DBConnException, 37
  - DBConnException, 38
- gldb::DBConnImp, 38
  - ~DBConnImp, 39
  - DBConnImp, 39
  - query, 39
  - select, 39
- gldb::DBConnMySQL, 40
  - ~DBConnMySQL, 41
  - DBConnMySQL, 41
  - m\_conn, 42
  - operator=, 41
  - query, 42
  - select, 42
- gldb::Table, 53

- ~Table, 54
- append\_record, 54
- create\_from\_file, 54
- get\_field, 55
- get\_headers, 55
- insert\_query, 55
- m\_headers, 56
- m\_quoted, 56
- m\_records, 56
- num\_fields, 55
- num\_records, 55
- set\_quoted, 56
- Table, 54
- gldb::TableBadInputFile, 56
  - TableBadInputFile, 57
- gldb::TableCouldNotOpenInputFile, 58
  - TableCouldNotOpenInputFile, 58
- gldb::TableException, 59
  - TableException, 59
- gldb::TableField, 60
  - ~TableField, 61
  - length, 61
  - m\_data, 63
  - operator std::string, 61
  - operator<<, 63
  - operator+=, 62
  - operator=, 62
  - TableField, 61
- gldb::TableMismatchedRecordLength, 63
  - TableMismatchedRecordLength, 64
- gldb::TableNoSuchField, 65
  - TableNoSuchField, 66
- gldb::TableNoSuchRecord, 66
  - TableNoSuchRecord, 67
- gldb::TableRow, 67
  - ~TableRow, 68
  - append\_field, 68
  - m\_fields, 70
  - print, 69
  - record\_string, 69
  - size, 70
  - TableRow, 68
- id
  - genleg::GLUser, 52
- insert\_query
  - gldb::Table, 55
- is\_set
  - genleg::Config, 24
- lastname
  - genleg::GLUser, 52
- length
  - gldb::TableField, 61
- lib/config/config.cpp, 71
- lib/config/config.h, 72
- lib/config/config\_getopt.cpp, 73
- lib/database/data\_structures.h, 74
- lib/database/database.h, 75
- lib/database/dbconn.cpp, 77
- lib/database/dbconn.h, 78
- lib/database/dbconnimp.h, 79
- lib/database/table.cpp, 81
- lib/database/table.h, 82
- lib/database/tablefield.cpp, 84
- lib/database/tablefield.h, 84
- lib/database/ablerow.cpp, 86
- lib/database/ablerow.h, 87
- lib/database\_imp/database\_imp.h, 88
- lib/database\_imp/dummy/dbconn\_dummy\_imp.cpp, 90
- lib/database\_imp/dummy/dbconn\_dummy\_imp.h, 91
- lib/database\_imp/mysql/dbconn\_mysql\_imp.cpp, 93
- lib/database\_imp/mysql/dbconn\_mysql\_imp.h, 94
- lib/dbsql/dbsql\_mysql.h, 96
- lib/dbsql/dbsqlstatements.cpp, 98
- lib/dbsql/dbsqlstatements.h, 99
- lib/gldb/gldatabase.cpp, 101
- lib/gldb/gldatabase.h, 101
- lib/gldb/gldb.h, 103
- lib/gldb/glexception.h, 104
- lib/gldb/gluser.cpp, 106
- lib/gldb/gluser.h, 106
- lib/stringhelp/stringhelp.cpp, 108
- lib/stringhelp/stringhelp.h, 108
- load\_sample\_data
  - genleg::GLDatabase, 49
- login
  - Database program., 21
  - gl\_user\_main.cpp, 115
  - Reporting program., 18
- m\_conn
  - gldb::DBConnMySQL, 42
- m\_data
  - gldb::TableField, 63
- m\_dbc
  - genleg::GLDatabase, 49
- m\_enabled
  - genleg::GLUser, 52
- m\_fields
  - gldb::TableRow, 70
- m\_firstname
  - genleg::GLUser, 52
- m\_headers
  - gldb::Table, 56
- m\_id
  - genleg::GLUser, 52
- m\_imp
  - gldb::DBConn, 33
- m\_lastname
  - genleg::GLUser, 52
- m\_opts\_set
  - genleg::Config, 25
- m\_opts\_supp
  - genleg::Config, 25
- m\_quoted
  - gldb::Table, 56
- m\_records

- gldb::Table, 56
- m\_sql
  - genleg::GLDatabase, 49
- m\_tables
  - genleg::GLDatabase, 49
- m\_username
  - genleg::GLUser, 52
- m\_views
  - genleg::GLDatabase, 49
- main
  - Database program., 21
  - gl\_user\_main.cpp, 115
  - Reporting program., 18
- num\_fields
  - gldb::Table, 55
- num\_records
  - gldb::Table, 55
- operator std::string
  - gldb::TableField, 61
- operator<<
  - gldb::TableField, 63
- operator+=
  - gldb::TableField, 62
- operator=
  - gldb::DBConn, 32
  - gldb::DBConnDummy, 37
  - gldb::DBConnMySQL, 41
  - gldb::TableField, 62
- populate\_from\_cmdline
  - genleg::Config, 24
- populate\_from\_file
  - genleg::Config, 25
- print
  - gldb::TableRow, 69
- Program configuration module, 15
- progs/gl\_db/gl\_db\_main.cpp, 110
- progs/gl\_report/gl\_report\_main.cpp, 111
- progs/gl\_user/gl\_user\_main.cpp, 113
- query
  - gldb::DBConn, 32
  - gldb::DBConnImp, 39
  - gldb::DBConnMySQL, 42
- record\_string
  - gldb::TableRow, 69
- Reporting program., 18
  - login, 18
  - main, 18
  - set\_configuration, 19
- SQL statements module, 14
- select
  - gldb::DBConn, 32
  - gldb::DBConnDummy, 37
  - gldb::DBConnImp, 39
  - gldb::DBConnMySQL, 42
- set\_configuration
  - Database program., 21
  - gl\_user\_main.cpp, 115
  - Reporting program., 19
- set\_quoted
  - gldb::Table, 56
- size
  - gldb::TableRow, 70
- split
  - General purpose helpers., 16
- Table
  - gldb::Table, 54
- TableBadInputFile
  - gldb::TableBadInputFile, 57
- TableCouldNotOpenInputFile
  - gldb::TableCouldNotOpenInputFile, 58
- TableException
  - gldb::TableException, 59
- TableField
  - gldb::TableField, 61
- TableMismatchedRecordLength
  - gldb::TableMismatchedRecordLength, 64
- TableNoSuchField
  - gldb::TableNoSuchField, 66
- TableNoSuchRecord
  - gldb::TableNoSuchRecord, 67
- TableRow
  - gldb::TableRow, 68
- trim
  - General purpose helpers., 16
- trim\_back
  - General purpose helpers., 17
- trim\_front
  - General purpose helpers., 17
- user\_by\_id
  - genleg::DBSQLStatements, 45
- user\_by\_username
  - genleg::DBSQLStatements, 46
- username
  - genleg::GLUser, 52