

general\_ledger

Generated by Doxygen 1.8.1.2

Fri Jun 13 2014 18:11:33



# Contents

<b>1</b>	<b>General Ledger.</b>	<b>1</b>
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Class Documentation</b>	<b>9</b>
5.1	genleg::Config Class Reference . . . . .	9
5.1.1	Detailed Description . . . . .	9
5.1.2	Constructor & Destructor Documentation . . . . .	9
5.1.2.1	Config . . . . .	9
5.1.2.2	~Config . . . . .	10
5.1.3	Member Function Documentation . . . . .	10
5.1.3.1	add_cmdline_option . . . . .	10
5.1.3.2	is_set . . . . .	10
5.1.3.3	operator[] . . . . .	10
5.1.3.4	populate_from_cmdline . . . . .	10
5.1.3.5	populate_from_file . . . . .	11
5.1.4	Member Data Documentation . . . . .	11
5.1.4.1	m_opts_set . . . . .	11
5.1.4.2	m_opts_supp . . . . .	11
5.2	genleg::ConfigBadConfigFile Class Reference . . . . .	11
5.2.1	Detailed Description . . . . .	11
5.3	genleg::ConfigBadOption Class Reference . . . . .	11
5.3.1	Detailed Description . . . . .	12
5.4	genleg::ConfigCouldNotOpenFile Class Reference . . . . .	12
5.4.1	Detailed Description . . . . .	12
5.5	genleg::ConfigOptionNotSet Class Reference . . . . .	12

5.5.1	Detailed Description	12
5.6	gldb::DBConn Class Reference	12
5.6.1	Detailed Description	13
5.6.2	Constructor & Destructor Documentation	13
5.6.2.1	DBConn	13
5.6.2.2	DBConn	13
5.6.3	Member Function Documentation	13
5.6.3.1	operator=	13
5.6.3.2	select	13
5.6.4	Member Data Documentation	14
5.6.4.1	m_imp	14
5.7	gldb::DBConnCouldNotConnect Class Reference	14
5.7.1	Detailed Description	14
5.7.2	Constructor & Destructor Documentation	14
5.7.2.1	DBConnCouldNotConnect	14
5.8	gldb::DBConnCouldNotQuery Class Reference	14
5.8.1	Detailed Description	14
5.8.2	Constructor & Destructor Documentation	15
5.8.2.1	DBConnCouldNotQuery	15
5.9	gldb::DBConnDummy Class Reference	15
5.9.1	Detailed Description	16
5.9.2	Constructor & Destructor Documentation	16
5.9.2.1	DBConnDummy	16
5.9.2.2	DBConnDummy	16
5.9.2.3	~DBConnDummy	16
5.9.3	Member Function Documentation	16
5.9.3.1	operator=	16
5.9.3.2	select	16
5.10	gldb::DBConnImp Class Reference	17
5.10.1	Detailed Description	17
5.10.2	Constructor & Destructor Documentation	17
5.10.2.1	DBConnImp	17
5.10.2.2	~DBConnImp	17
5.10.3	Member Function Documentation	17
5.10.3.1	select	17
5.11	gldb::DBConnMySQL Class Reference	18
5.11.1	Detailed Description	19
5.11.2	Constructor & Destructor Documentation	19
5.11.2.1	DBConnMySQL	19
5.11.2.2	DBConnMySQL	19

5.11.2.3	<a href="#">~DBConnMySQL</a>	19
5.11.3	<a href="#">Member Function Documentation</a>	19
5.11.3.1	<a href="#">operator=</a>	19
5.11.3.2	<a href="#">select</a>	19
5.11.4	<a href="#">Member Data Documentation</a>	20
5.11.4.1	<a href="#">m_conn</a>	20
5.12	<a href="#">gldb::Table Class Reference</a>	20
5.12.1	<a href="#">Detailed Description</a>	21
5.12.2	<a href="#">Constructor &amp; Destructor Documentation</a>	21
5.12.2.1	<a href="#">Table</a>	21
5.12.2.2	<a href="#">~Table</a>	21
5.12.3	<a href="#">Member Function Documentation</a>	21
5.12.3.1	<a href="#">append_record</a>	21
5.12.3.2	<a href="#">get_headers</a>	21
5.12.3.3	<a href="#">num_fields</a>	22
5.12.3.4	<a href="#">num_records</a>	22
5.12.3.5	<a href="#">operator[]</a>	22
5.12.4	<a href="#">Member Data Documentation</a>	22
5.12.4.1	<a href="#">m_headers</a>	22
5.12.4.2	<a href="#">m_records</a>	22
5.13	<a href="#">gldb::TableField Class Reference</a>	22
5.13.1	<a href="#">Detailed Description</a>	24
5.13.2	<a href="#">Constructor &amp; Destructor Documentation</a>	24
5.13.2.1	<a href="#">TableField</a>	24
5.13.2.2	<a href="#">TableField</a>	24
5.13.2.3	<a href="#">~TableField</a>	24
5.13.3	<a href="#">Member Function Documentation</a>	24
5.13.3.1	<a href="#">length</a>	24
5.13.3.2	<a href="#">operator std::string</a>	24
5.13.3.3	<a href="#">operator+=</a>	24
5.13.3.4	<a href="#">operator+=</a>	25
5.13.3.5	<a href="#">operator=</a>	25
5.13.3.6	<a href="#">operator=</a>	25
5.13.3.7	<a href="#">operator[]</a>	25
5.13.3.8	<a href="#">operator[]</a>	26
5.13.4	<a href="#">Friends And Related Function Documentation</a>	26
5.13.4.1	<a href="#">operator&lt;&lt;</a>	26
5.13.5	<a href="#">Member Data Documentation</a>	26
5.13.5.1	<a href="#">m_data</a>	26
5.14	<a href="#">gldb::TableRow Class Reference</a>	26

5.14.1	Detailed Description	27
5.14.2	Constructor & Destructor Documentation	27
5.14.2.1	TableRow	27
5.14.2.2	TableRow	27
5.14.2.3	~TableRow	27
5.14.3	Member Function Documentation	27
5.14.3.1	append_field	27
5.14.3.2	append_field	27
5.14.3.3	append_field	28
5.14.3.4	operator[]	28
5.14.3.5	operator[]	28
5.14.3.6	print	28
5.14.3.7	size	28
5.14.4	Member Data Documentation	29
5.14.4.1	m_fields	29
<b>6</b>	<b>File Documentation</b>	<b>31</b>
6.1	lib/config/config.cpp File Reference	31
6.1.1	Detailed Description	31
6.2	lib/config/config.h File Reference	32
6.2.1	Detailed Description	33
6.3	lib/config/config_getopt.cpp File Reference	33
6.3.1	Detailed Description	33
6.3.2	Macro Definition Documentation	34
6.3.2.1	_XOPEN_SOURCE	34
6.4	lib/database/data_structures.h File Reference	34
6.4.1	Detailed Description	35
6.5	lib/database/database.h File Reference	36
6.5.1	Detailed Description	37
6.6	lib/database/dbconn.cpp File Reference	37
6.6.1	Detailed Description	38
6.7	lib/database/dbconn.h File Reference	39
6.7.1	Detailed Description	40
6.8	lib/database/dbconnimp.h File Reference	40
6.8.1	Detailed Description	42
6.9	lib/database/table.cpp File Reference	43
6.9.1	Detailed Description	43
6.10	lib/database/table.h File Reference	44
6.10.1	Detailed Description	45
6.11	lib/database/tablefield.cpp File Reference	45

6.11.1 Detailed Description . . . . .	46
6.12 lib/database/tablefield.h File Reference . . . . .	46
6.12.1 Detailed Description . . . . .	48
6.13 lib/database/tablerow.cpp File Reference . . . . .	48
6.13.1 Detailed Description . . . . .	48
6.14 lib/database/tablerow.h File Reference . . . . .	49
6.14.1 Detailed Description . . . . .	50
6.15 lib/database_imp/database_imp.h File Reference . . . . .	50
6.15.1 Detailed Description . . . . .	51
6.16 lib/database_imp/dummy/dbconn_dummy_imp.cpp File Reference . . . . .	52
6.16.1 Detailed Description . . . . .	53
6.17 lib/database_imp/dummy/dbconn_dummy_imp.h File Reference . . . . .	54
6.17.1 Detailed Description . . . . .	55
6.18 lib/database_imp/mysql/dbconn_mysql_imp.cpp File Reference . . . . .	55
6.18.1 Detailed Description . . . . .	56
6.19 lib/database_imp/mysql/dbconn_mysql_imp.h File Reference . . . . .	56
6.19.1 Detailed Description . . . . .	57
6.20 lib/stringhelp/stringhelp.cpp File Reference . . . . .	58
6.20.1 Detailed Description . . . . .	58
6.21 lib/stringhelp/stringhelp.h File Reference . . . . .	58
6.21.1 Detailed Description . . . . .	59





## 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

# Class Index

### 2.1 Class Hierarchy

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

genleg::Config . . . . .	9
genleg::ConfigBadConfigFile . . . . .	11
genleg::ConfigBadOption . . . . .	11
genleg::ConfigCouldNotOpenFile . . . . .	12
genleg::ConfigOptionNotSet . . . . .	12
gldb::DBConn . . . . .	12
gldb::DBConnCouldNotConnect . . . . .	14
gldb::DBConnCouldNotQuery . . . . .	14
gldb::DBConnImp . . . . .	17
gldb::DBConnDummy . . . . .	15
gldb::DBConnMySQL . . . . .	18
gldb::Table . . . . .	20
gldb::TableField . . . . .	22
gldb::TableRow . . . . .	26



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">genleg::Config</a>	9
<a href="#">genleg::ConfigBadConfigFile</a>	11
<a href="#">genleg::ConfigBadOption</a>	11
<a href="#">genleg::ConfigCouldNotOpenFile</a>	12
<a href="#">genleg::ConfigOptionNotSet</a>	12
<a href="#">gldb::DBConn</a>	12
<a href="#">gldb::DBConnCouldNotConnect</a>	14
<a href="#">gldb::DBConnCouldNotQuery</a>	14
<a href="#">gldb::DBConnDummy</a>	15
<a href="#">gldb::DBConnImp</a>	17
<a href="#">gldb::DBConnMySQL</a>	18
<a href="#">gldb::Table</a>	20
<a href="#">gldb::TableField</a>	22
<a href="#">gldb::TableRow</a>	26



## Chapter 4

# File Index

### 4.1 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 . . . . .	31
lib/config/ <a href="#">config.h</a>	
Interface to program configurations class . . . . .	32
lib/config/ <a href="#">config_getopt.cpp</a>	
Implementation of command line functionality . . . . .	33
lib/database/ <a href="#">data_structures.h</a>	
Main interface to database data structures . . . . .	34
lib/database/ <a href="#">database.h</a>	
User interface to database functionality . . . . .	36
lib/database/ <a href="#">dbconn.cpp</a>	
Implementation of database connection class . . . . .	37
lib/database/ <a href="#">dbconn.h</a>	
Interface to database connection base class . . . . .	39
lib/database/ <a href="#">dbconnimp.h</a>	
Interface to abstract database implementation base class . . . . .	40
lib/database/ <a href="#">table.cpp</a>	
Implementation of database table data structure . . . . .	43
lib/database/ <a href="#">table.h</a>	
Interface to database table data structure . . . . .	44
lib/database/ <a href="#">tablefield.cpp</a>	
Implementation of database table field class . . . . .	45
lib/database/ <a href="#">tablefield.h</a>	
Interface to database table field class . . . . .	46
lib/database/ <a href="#">tablerow.cpp</a>	
Implementation of database table row data structure . . . . .	48
lib/database/ <a href="#">tablerow.h</a>	
Interface to database table row data structure . . . . .	49
lib/database_imp/ <a href="#">database_imp.h</a>	
Interface to database implementation factory function . . . . .	50
lib/database_imp/dummy/ <a href="#">dbconn_dummy_imp.cpp</a>	
Implementation of Dummy database connection implementation class . . . . .	52
lib/database_imp/dummy/ <a href="#">dbconn_dummy_imp.h</a>	
Interface to dummy database connection implementation class . . . . .	54
lib/database_imp/mysql/ <a href="#">dbconn_mysql_imp.cpp</a>	
Implementation of MySQL database connection implementation class . . . . .	55
lib/database_imp/mysql/ <a href="#">dbconn_mysql_imp.h</a>	
Interface to MySQL database connection implementation class . . . . .	56

lib/stringhelp/ <a href="#">stringhelp.cpp</a>	
Implementation of string helper functions . . . . .	58
lib/stringhelp/ <a href="#">stringhelp.h</a>	
Interface to string helper functions . . . . .	58



## Chapter 5

# Class Documentation

### 5.1 genleg::Config Class Reference

```
#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](#)

#### 5.1.1 Detailed Description

Configuration options class

#### 5.1.2 Constructor & Destructor Documentation

##### 5.1.2.1 Config::Config ( )

Constructor

### 5.1.2.2 Config::~Config ( )

Destructor

## 5.1.3 Member Function Documentation

### 5.1.3.1 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.

### 5.1.3.2 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.

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

operator[] overload.

Retrieves the value of a set option.

#### Parameters

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

#### Returns

The value of the option.

#### Exceptions

<i>ConfigOptionNotSet()</i>	if the named option has not been set.
-----------------------------	---------------------------------------

### 5.1.3.4 void Config::populate\_from\_cmdline ( const int *argc*, char \*const \* *argv* )

Populates options from the command line.

#### Parameters

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

## Exceptions

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

## 5.1.3.5 void Config::populate\_from\_file ( const std::string filename )

Populates options from a configuration file.

## Parameters

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

## Exceptions

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

## 5.1.4 Member Data Documentation

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

Map of options which have been set

## 5.1.4.2 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](#)

## 5.2 genleg::ConfigBadConfigFile Class Reference

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

## 5.2.1 Detailed Description

Exception class for badly formed configuration file

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

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

## 5.3 genleg::ConfigBadOption Class Reference

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

### 5.3.1 Detailed Description

Exception class for bad provided option

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

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

## 5.4 genleg::ConfigCouldNotOpenFile Class Reference

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

### 5.4.1 Detailed Description

Exception class for when conf file cannot be opened

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

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

## 5.5 genleg::ConfigOptionNotSet Class Reference

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

### 5.5.1 Detailed Description

Exception class for option not set

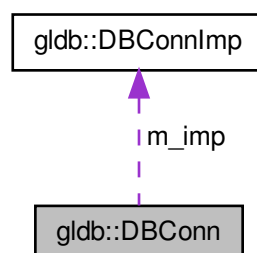
The documentation for this class was generated from the following file:

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

## 5.6 glldb::DBConn Class Reference

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

Collaboration diagram for glldb::DBConn:



## Public Member Functions

- [DBConn](#) ([DBConnImp](#) \*imp)  
*Constructor.*
- [~DBConn](#) ()  
*Destructor..*
- [Table select](#) (std::string query)  
*Runs an SQL SELECT query.*
- [DBConn](#) (const [DBConn](#) &)
- [DBConn](#) & [operator=](#) (const [DBConn](#) &)

## Private Attributes

- [DBConnImp](#) \* [m\\_imp](#)

### 5.6.1 Detailed Description

Database connection class

### 5.6.2 Constructor & Destructor Documentation

#### 5.6.2.1 [DBConn::DBConn](#) ( [DBConnImp](#) \* *imp* ) [explicit]

Constructor.

##### Parameters

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

#### 5.6.2.2 [gldb::DBConn::DBConn](#) ( const [DBConn](#) & )

Deleted copy constructor

### 5.6.3 Member Function Documentation

#### 5.6.3.1 [DBConn& gldb::DBConn::operator=](#) ( const [DBConn](#) & )

Deleted assignment operator

#### 5.6.3.2 [Table](#) [DBConn::select](#) ( std::string *query* )

Runs an SQL SELECT query.

##### Parameters

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

##### Returns

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

## 5.6.4 Member Data Documentation

### 5.6.4.1 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](#)

## 5.7 glldb::DBConnCouldNotConnect Class Reference

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

### Public Member Functions

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

### 5.7.1 Detailed Description

Could not connect to database exception class

### 5.7.2 Constructor & Destructor Documentation

#### 5.7.2.1 glldb::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](#)

## 5.8 glldb::DBConnCouldNotQuery Class Reference

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

### Public Member Functions

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

### 5.8.1 Detailed Description

Could not execute database query exception class

## 5.8.2 Constructor & Destructor Documentation

### 5.8.2.1 glldb::DBConnCouldNotQuery::DBConnCouldNotQuery ( const std::string & msg ) [inline], [explicit]

Constructor.

Parameters

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

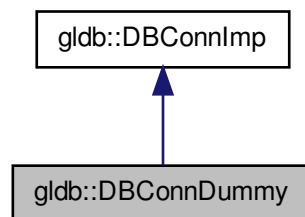
The documentation for this class was generated from the following file:

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

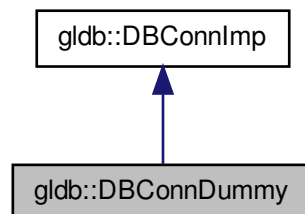
## 5.9 glldb::DBConnDummy Class Reference

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

Inheritance diagram for glldb::DBConnDummy:



Collaboration diagram for glldb::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.*

### 5.9.1 Detailed Description

Dummy database implementation class

### 5.9.2 Constructor & Destructor Documentation

#### 5.9.2.1 [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.

#### 5.9.2.2 [gldb::DBConnDummy::DBConnDummy](#) ( const [DBConnDummy](#) & )

Deleted copy constructor

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

Destructor

### 5.9.3 Member Function Documentation

#### 5.9.3.1 [DBConnDummy& gldb::DBConnDummy::operator=](#) ( const [DBConnDummy](#) & )

Deleted assignment operator

#### 5.9.3.2 [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](#).



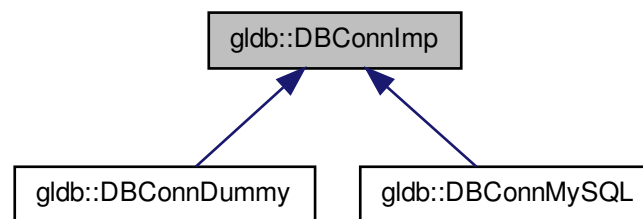
The documentation for this class was generated from the following files:

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

## 5.10 gldb::DBConnImp Class Reference

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

Inheritance diagram for gldb::DBConnImp:



### Public Member Functions

- [DBConnImp](#) ()
- virtual [~DBConnImp](#) ()
- virtual [Table select](#) (std::string query)=0  
*Runs an SQL SELECT query.*

#### 5.10.1 Detailed Description

Abstract database implementation base class

#### 5.10.2 Constructor & Destructor Documentation

5.10.2.1 `gldb::DBConnImp::DBConnImp ( ) [inline]`

Constructor

5.10.2.2 `virtual gldb::DBConnImp::~~DBConnImp ( ) [inline],[virtual]`

Destructor

#### 5.10.3 Member Function Documentation

5.10.3.1 `virtual Table gldb::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 [gldb::DBConnMySQL](#), and [gldb::DBConnDummy](#).

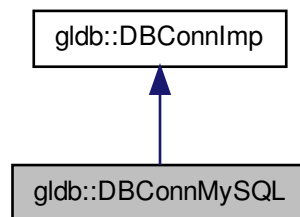
The documentation for this class was generated from the following file:

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

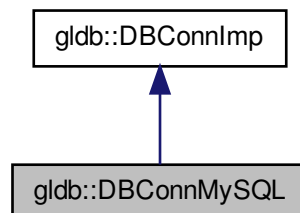
## 5.11 gldb::DBConnMySQL Class Reference

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

Inheritance diagram for `gldb::DBConnMySQL`:



Collaboration diagram for `gldb::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](#) &)
- [Table select](#) (std::string query)

*Runs an SQL SELECT query.*

## Private Attributes

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

### 5.11.1 Detailed Description

MySQL database implementation class

### 5.11.2 Constructor & Destructor Documentation

- 5.11.2.1 [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.
---	-----------------------------------

- 5.11.2.2 [gdb::DBConnMySQL::DBConnMySQL](#) ( const [DBConnMySQL](#) & )

Deleted copy constructor

- 5.11.2.3 [DBConnMySQL::~~DBConnMySQL](#) ( ) [\[virtual\]](#)

Destructor

### 5.11.3 Member Function Documentation

- 5.11.3.1 [DBConnMySQL& gdb::DBConnMySQL::operator=](#) ( const [DBConnMySQL](#) & )

Deleted assignment operator

- 5.11.3.2 [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 [gldb::DBConnImp](#).

### 5.11.4 Member Data Documentation

#### 5.11.4.1 `MYSQL* gldb::DBConnMySQL::m_conn` [private]

The initialized MySQL handle.

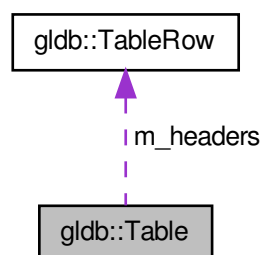
The documentation for this class was generated from the following files:

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

## 5.12 gldb::Table Class Reference

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

Collaboration diagram for `gldb::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.*
- `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.*

### Private Attributes

- `TableRow m_headers`
- `std::vector< TableRow > m_records`

## 5.12.1 Detailed Description

Database table class

## 5.12.2 Constructor & Destructor Documentation

### 5.12.2.1 Table::Table ( const TableRow & headers ) [explicit]

Constructor.

Parameters

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

### 5.12.2.2 Table::~~Table ( )

Destructor

## 5.12.3 Member Function Documentation

### 5.12.3.1 void Table::append\_record ( const TableRow & new\_record )

Appends a record to the table.

Parameters

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

### 5.12.3.2 const TableRow & Table::get\_headers ( ) const

Returns the field names.

Returns

The field names.

#### 5.12.3.3 `size_t Table::num_fields ( ) const`

Returns the number of fields in each row.

##### Returns

The number of fields in each row.

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

Returns the number of record in the table.

##### Returns

The number of records in the table.

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

Overloaded index operator.

##### Parameters

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

##### Returns

The selected record.

### 5.12.4 Member Data Documentation

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

The names of the fields

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

A vector of the records

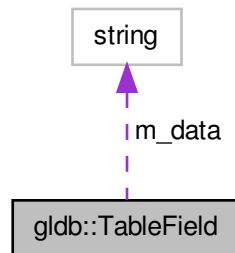
The documentation for this class was generated from the following files:

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

## 5.13 `glldb::TableField` Class Reference

```
#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.*

### 5.13.1 Detailed Description

Database table field class

### 5.13.2 Constructor & Destructor Documentation

#### 5.13.2.1 `TableField::TableField ( const char * data )` `[explicit]`

Constructor accepting `const char * data`.

##### Parameters

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

#### 5.13.2.2 `TableField::TableField ( const std::string & data )` `[explicit]`

Constructor accepting `std::string data`.

##### Parameters

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

#### 5.13.2.3 `TableField::~~TableField ( )`

Destructor

### 5.13.3 Member Function Documentation

#### 5.13.3.1 `size_t TableField::length ( ) const`

Returns the length of the field.

##### Returns

The length of the field.

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

Overridden conversion operator.

Returns the field contents as a string.

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

Overridden compound assignment operator.

##### Parameters

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



**Returns**

A reference to the same field.

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

Overridden compound assignment operator.

**Parameters**

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

**Returns**

A reference to the same field.

**5.13.3.5 TableField & TableField::operator= ( const char \* *data* )**

Overridden assignment operator for `const char *`.

**Parameters**

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

**Returns**

A reference to the same field.

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

Overridden assignment operator for `std::string`.

**Parameters**

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

**Returns**

A reference to the same field.

**5.13.3.7 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.

#### 5.13.3.8 `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.

### 5.13.4 Friends And Related Function Documentation

#### 5.13.4.1 `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`.

### 5.13.5 Member Data Documentation

#### 5.13.5.1 `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](#)

## 5.14 `glldb::TableRow` Class Reference

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

### Public Member Functions

- [TableRow](#) ()
- [TableRow](#) (const size\_t *size*)  
*Constructor with initial number of fields.*
- [~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.*

## Private Attributes

- std::vector< [TableField](#) > [m\\_fields](#)

### 5.14.1 Detailed Description

Database table row class

### 5.14.2 Constructor & Destructor Documentation

#### 5.14.2.1 TableRow::TableRow ( )

Default constructor

#### 5.14.2.2 TableRow::TableRow ( const size\_t size ) [explicit]

Constructor with initial number of fields.

##### Parameters

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

#### 5.14.2.3 TableRow::~~TableRow ( )

Destructor

### 5.14.3 Member Function Documentation

#### 5.14.3.1 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.
------------------	--------------------------------

#### 5.14.3.2 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.
------------------	--------------------------------

**5.14.3.3 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.
------------------	-----------------------------

**5.14.3.4 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.

**5.14.3.5 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.

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

Prints a row.

## Parameters

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

**5.14.3.7 size\_t TableRow::size ( ) const**

Returns the number of fields.

## Returns

The number of fields.

### 5.14.4 Member Data Documentation

#### 5.14.4.1 `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](#)



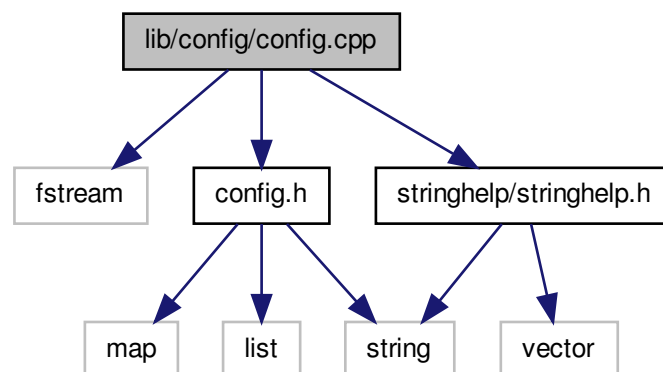
## Chapter 6

# File Documentation

### 6.1 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:
```



#### 6.1.1 Detailed Description

Implementation of program configurations class.

##### Author

Paul Griffiths

##### Copyright

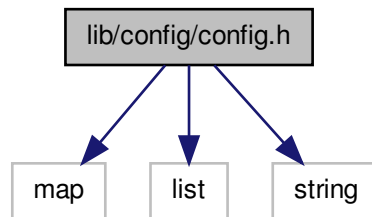
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

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

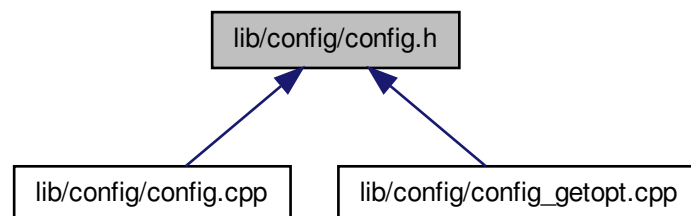
Interface to program configurations class.

```
#include <map>
#include <list>
#include <string>
```

Include dependency graph for config.h:



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



### Classes

- class `genleg::ConfigOptionNotSet`
- class `genleg::ConfigBadOption`
- class `genleg::ConfigCouldNotOpenFile`
- class `genleg::ConfigBadConfigFile`
- class `genleg::Config`

### Enumerations

- enum **Argument**



### 6.2.1 Detailed Description

Interface to program configurations class.

#### Author

Paul Griffiths

#### Copyright

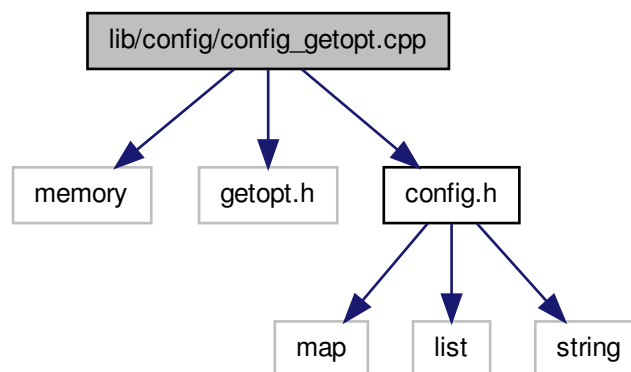
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 6.3 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`

### 6.3.1 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/>

## 6.3.2 Macro Definition Documentation

### 6.3.2.1 `#define _XOPEN_SOURCE 600`

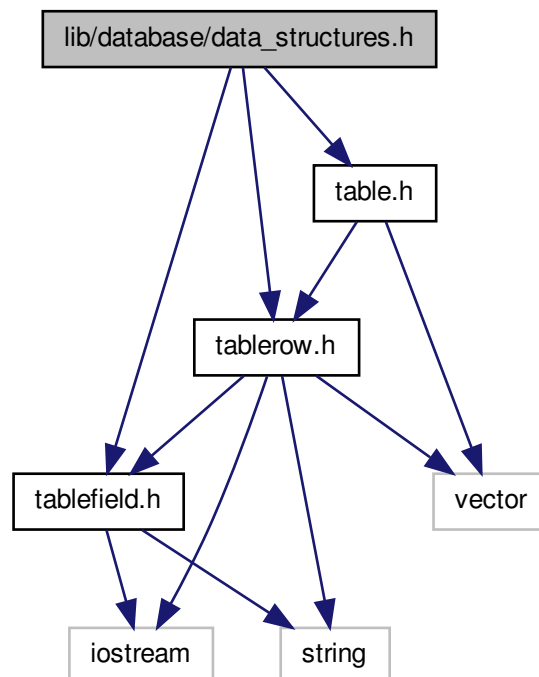
UNIX feature test macro for getopt library

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

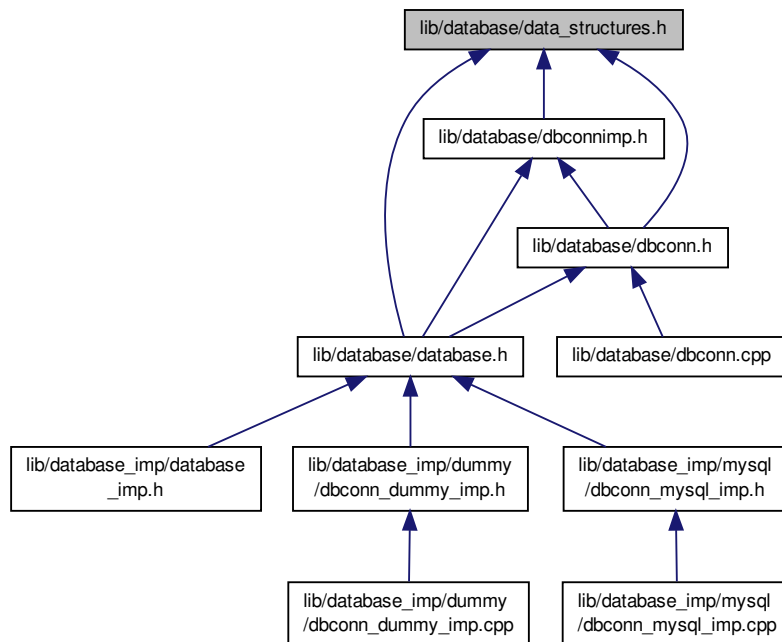
Main interface to database data structures.

```
#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:



### 6.4.1 Detailed Description

Main interface to database data structures.

#### Author

Paul Griffiths

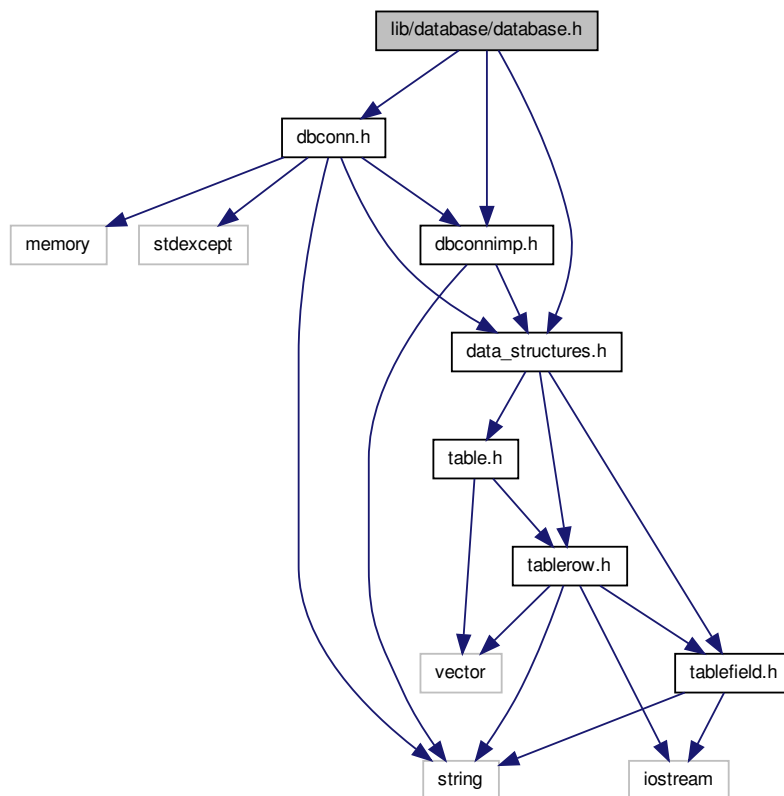
## Copyright

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

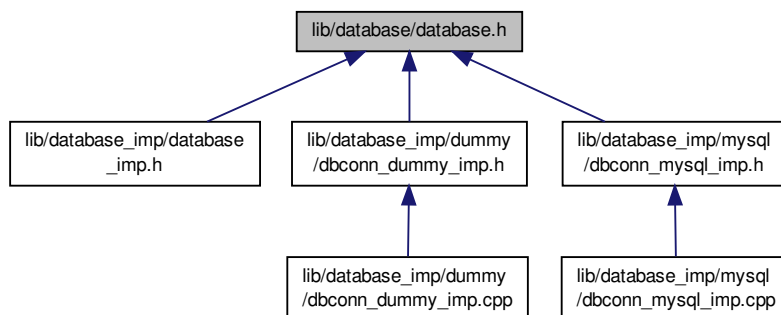
## 6.5 lib/database/database.h File Reference

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:



### 6.5.1 Detailed Description

User interface to database functionality.

#### Author

Paul Griffiths

#### Copyright

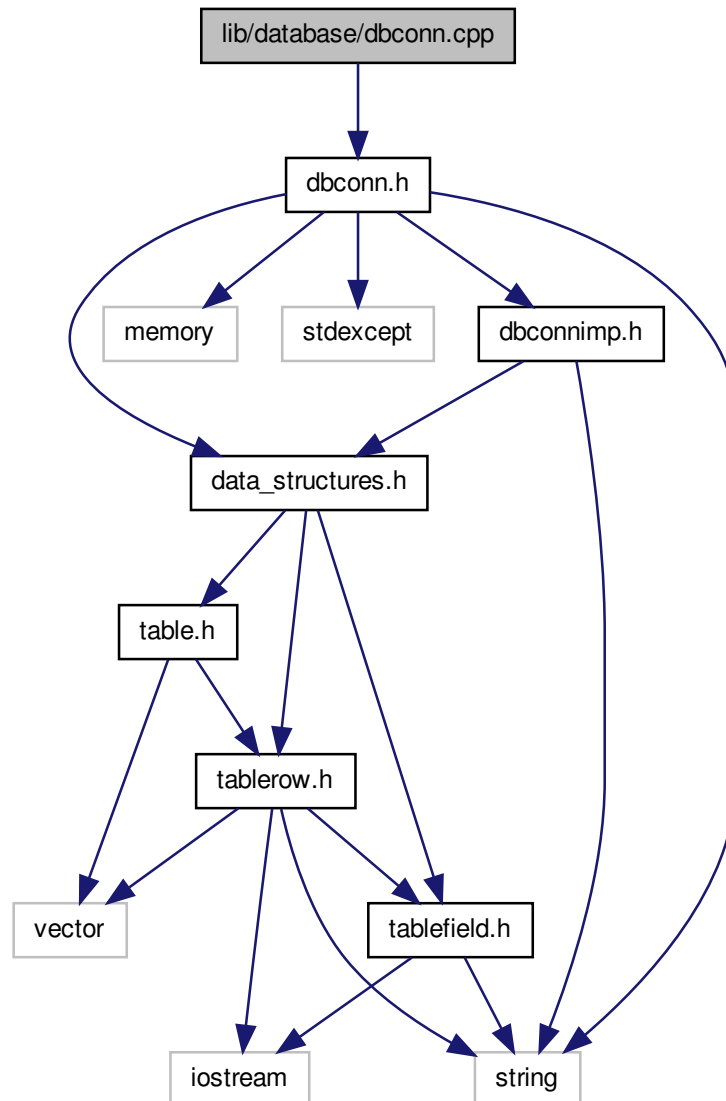
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

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

Implementation of database connection class.

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

Include dependency graph for dbconn.cpp:



### 6.6.1 Detailed Description

Implementation of database connection class.

Author

Paul Griffiths

Copyright

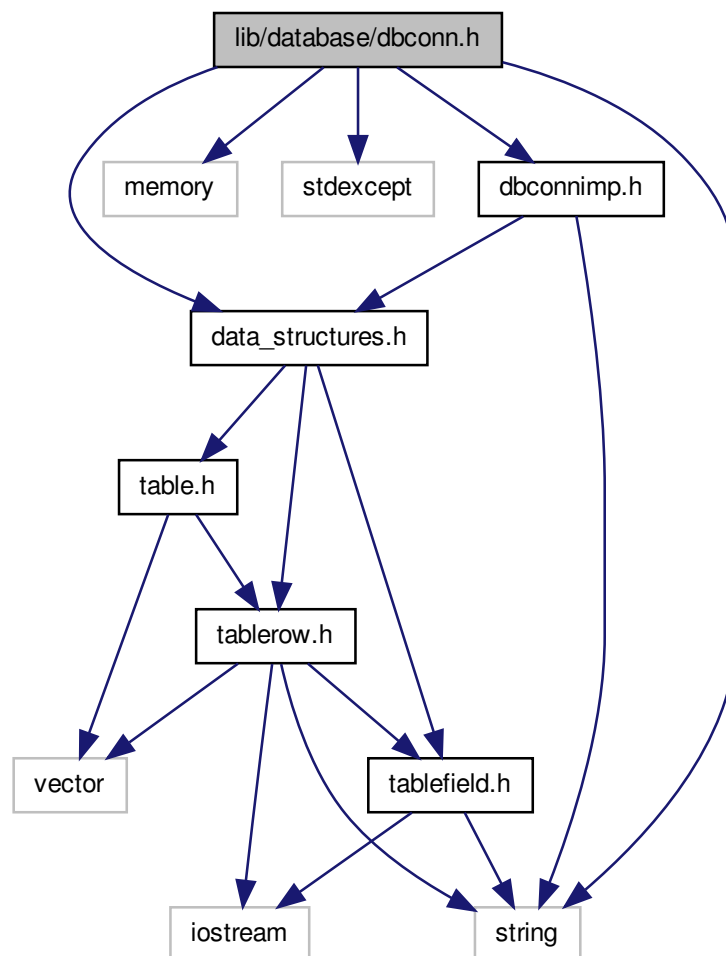
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

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

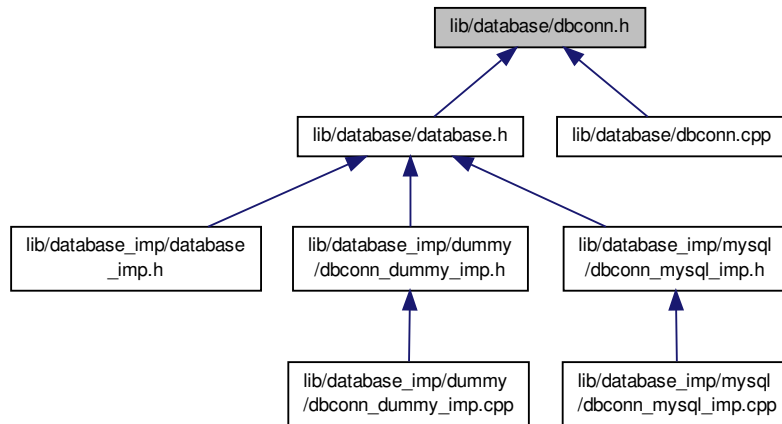
Interface to database connection base class.

```
#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::DBConnCouldNotConnect](#)
- class [gldb::DBConnCouldNotQuery](#)
- class [gldb::DBConn](#)

### 6.7.1 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/>

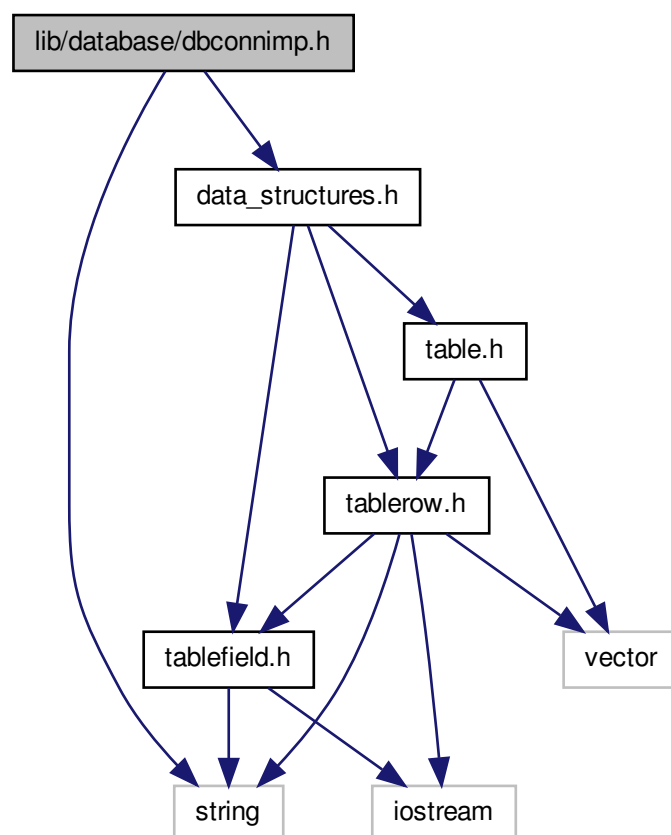
## 6.8 lib/database/dbconnimp.h File Reference

Interface to abstract database implementation base class.

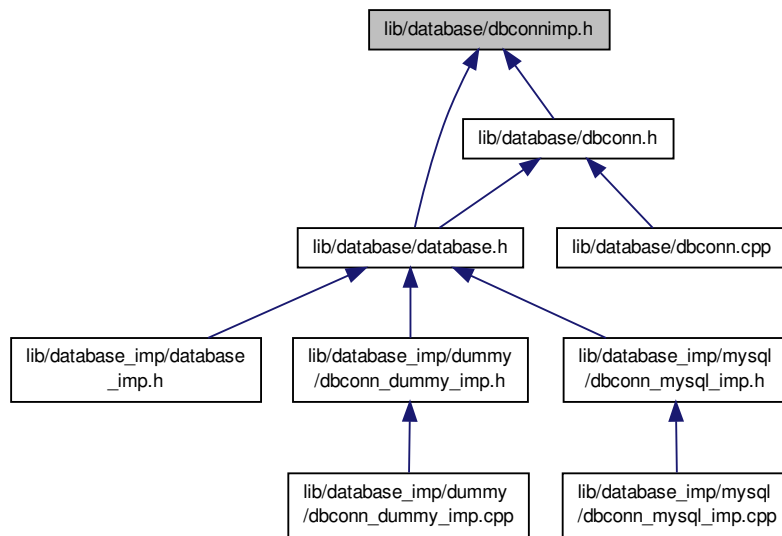
```
#include <string>
#include "data_structures.h"
```



Include dependency graph for dbconnimp.h:



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



## Classes

- class [gldb::DBConnImp](#)

### 6.8.1 Detailed Description

Interface to abstract database implementation base class.

#### Author

Paul Griffiths

### Copyright

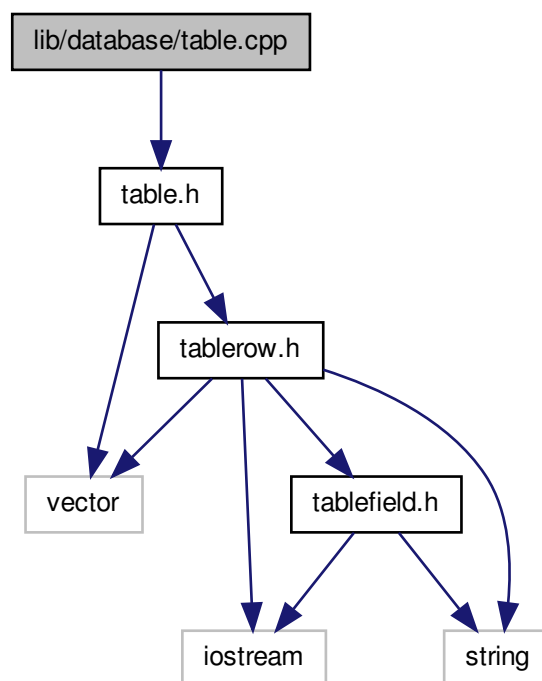
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

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

Implementation of database table data structure.

```
#include "table.h"
```

Include dependency graph for table.cpp:



### 6.9.1 Detailed Description

Implementation of database table data structure.

#### Author

Paul Griffiths

### Copyright

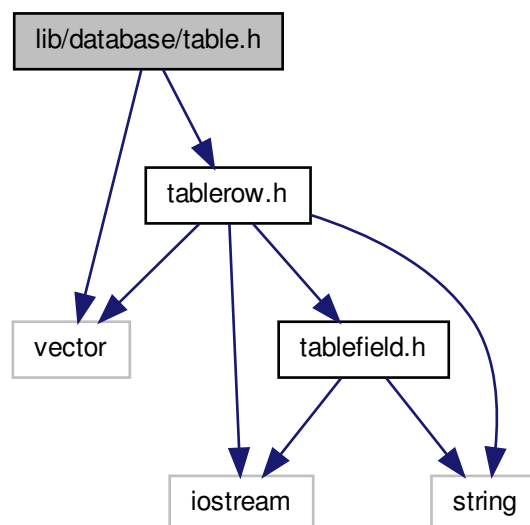
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

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

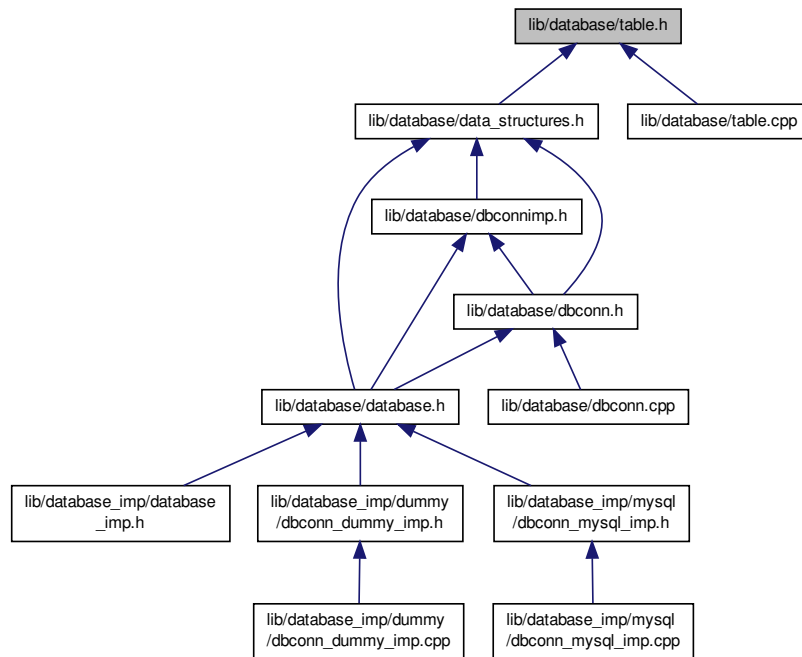
Interface to database table data structure.

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

Include dependency graph for table.h:



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



## Classes

- class [gldb::Table](#)

### 6.10.1 Detailed Description

Interface to database table data structure.

#### Author

Paul Griffiths

#### Copyright

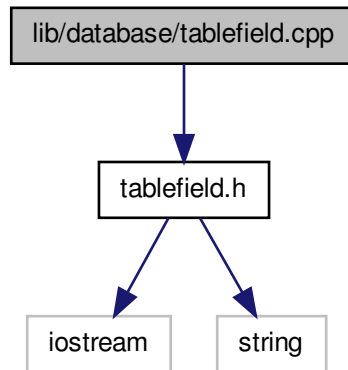
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

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

Implementation of database table field class.

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

Include dependency graph for tablefield.cpp:



### 6.11.1 Detailed Description

Implementation of database table field class.

#### Author

Paul Griffiths

#### Copyright

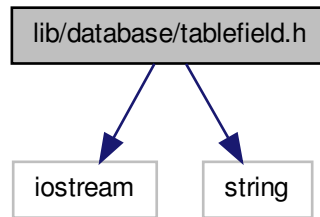
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

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

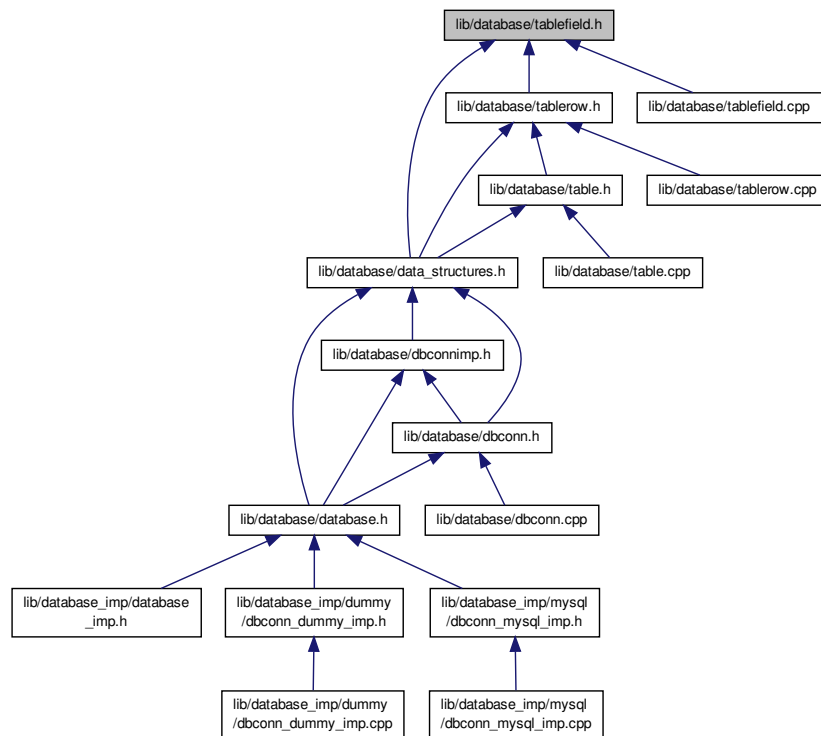
Interface to database table field class.

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

Include dependency graph for tablefield.h:



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



## Classes

- class [gldb::TableField](#)

## Functions

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

### 6.12.1 Detailed Description

Interface to database table field class.

**Author**

Paul Griffiths

**Copyright**

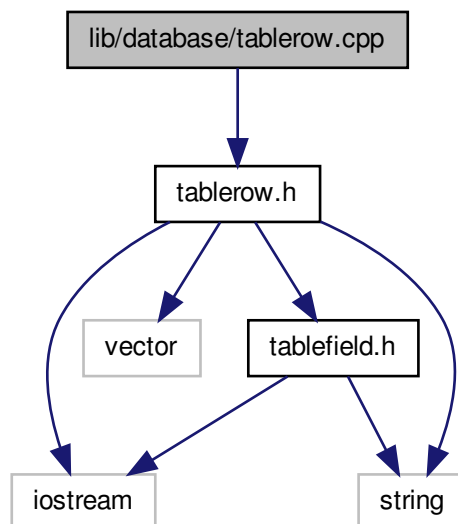
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

### 6.13 lib/database/tablerow.cpp File Reference

Implementation of database table row data structure.

```
#include "tablerow.h"
```

Include dependency graph for tablerow.cpp:



### 6.13.1 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/>

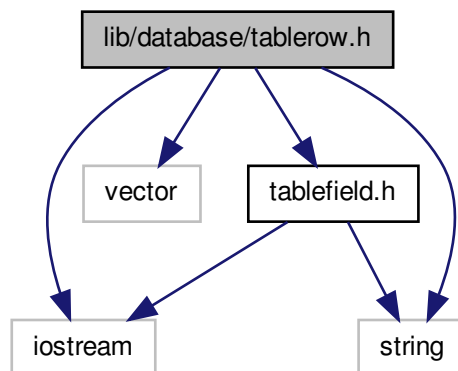


## 6.14 lib/database/tablerow.h File Reference

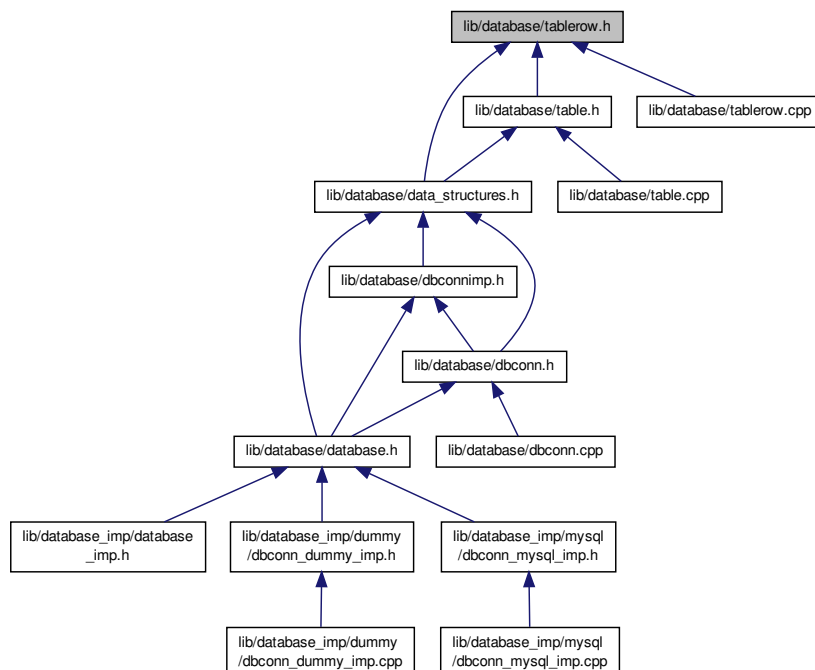
Interface to database table row data structure.

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

Include dependency graph for tablerow.h:



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



## Classes

- class `gldb::TableRow`

### 6.14.1 Detailed Description

Interface to database table row data structure.

#### Author

Paul Griffiths

#### Copyright

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

## 6.15 lib/database\_imp/database\_imp.h File Reference

Interface to database implementation factory function.

```
#include <memory>
#include "database/database.h"
```



**Author**

Paul Griffiths

**Copyright**

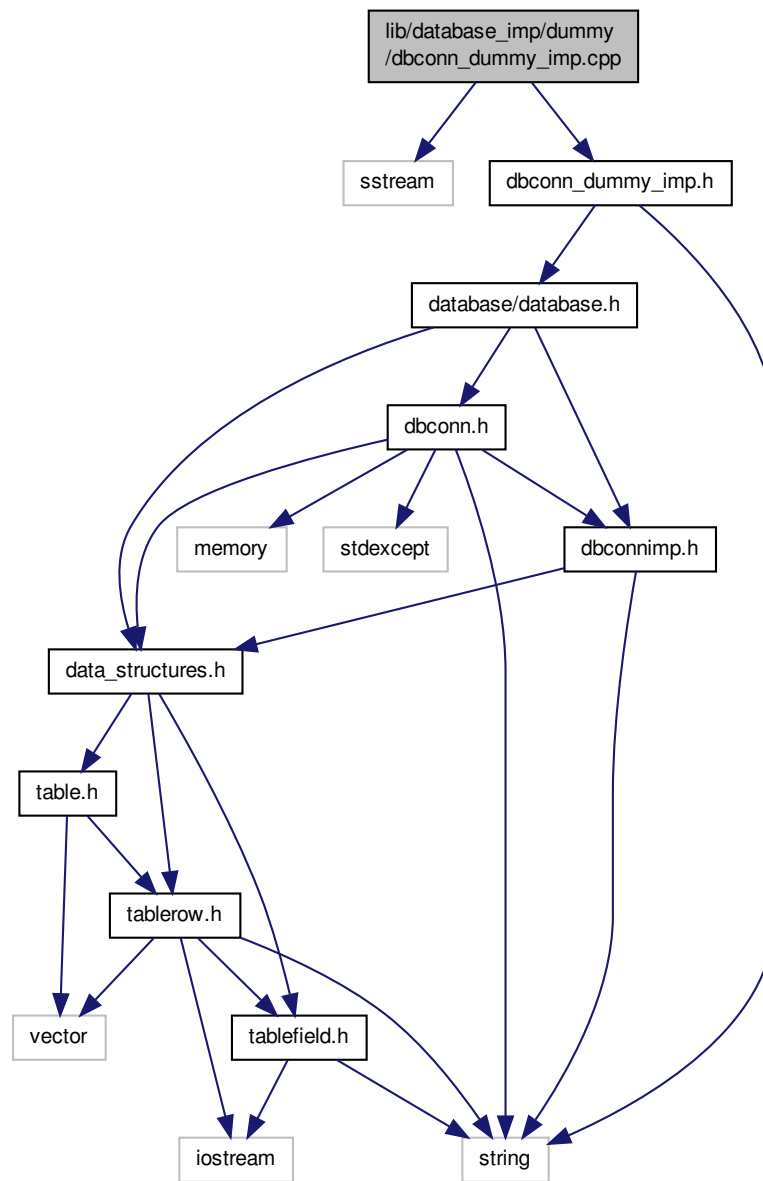
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

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

Implementation of Dummy database connection implementation class.

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

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



### 6.16.1 Detailed Description

Implementation of Dummy database connection implementation class.

Author

Paul Griffiths

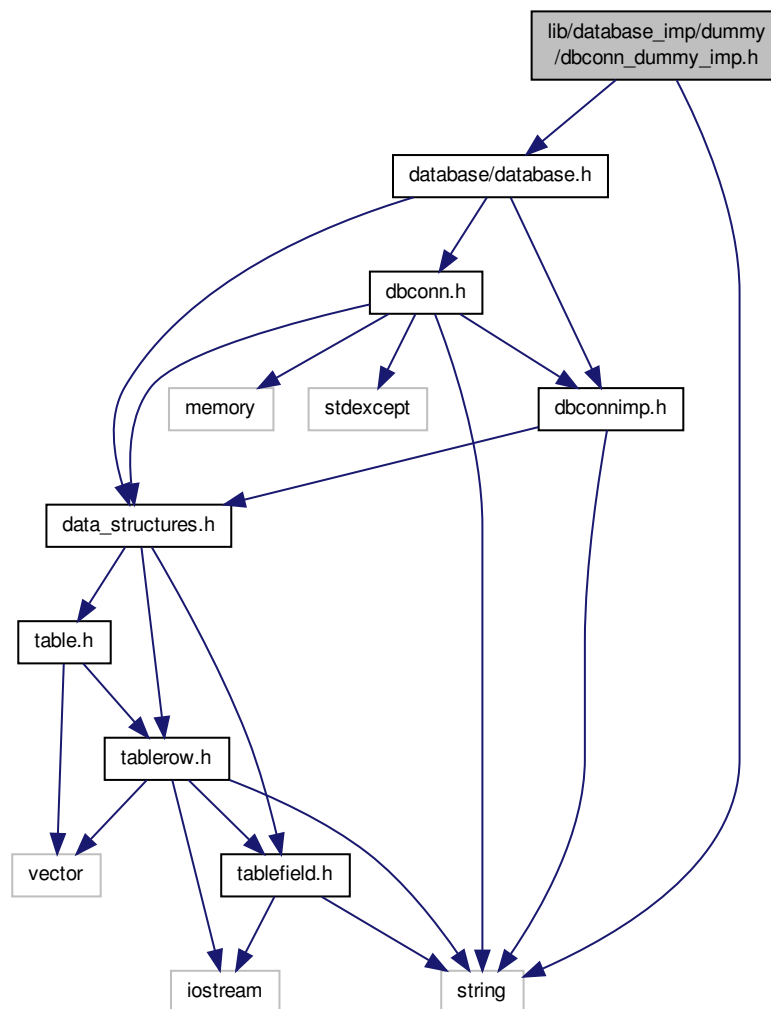
Copyright

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

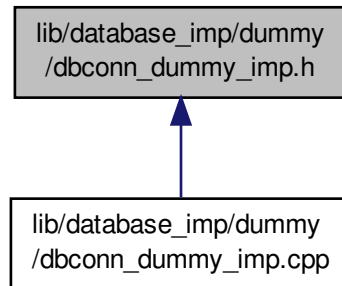
## 6.17 lib/database\_imp/dummy/dbconn\_dummy\_imp.h File Reference

Interface to dummy database connection implementation class.

```
#include <string>
#include "database/database.h"
Include dependency graph for dbconn_dummy_imp.h:
```



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



## Classes

- class [gldb::DBConnDummy](#)

### 6.17.1 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/>

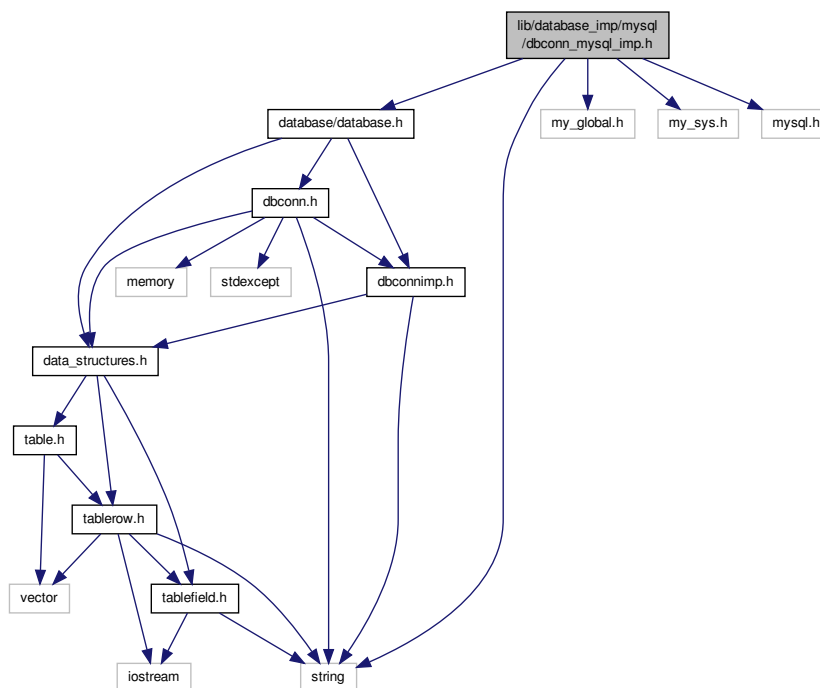
## 6.18 lib/database\_imp/mysql/dbconn\_mysql\_imp.cpp File Reference

Implementation of MySQL database connection implementation class.

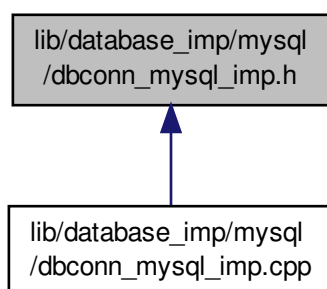




Include dependency graph for dbconn\_mysql\_imp.h:



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



## Classes

- class [gldb::DBConnMySQL](#)

## 6.19.1 Detailed Description

Interface to MySQL database connection implementation class.

**Author**

Paul Griffiths

**Copyright**

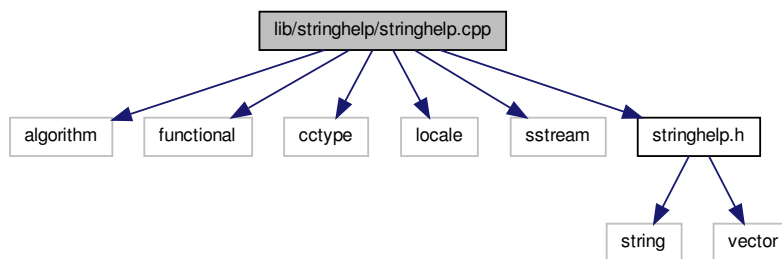
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

## 6.20 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:



### 6.20.1 Detailed Description

Implementation of string helper functions.

**Author**

Paul Griffiths

**Copyright**

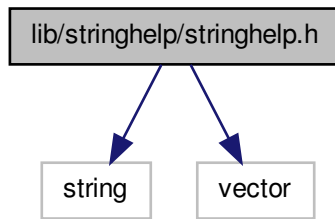
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. <http://www.gnu.org/licenses/>

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

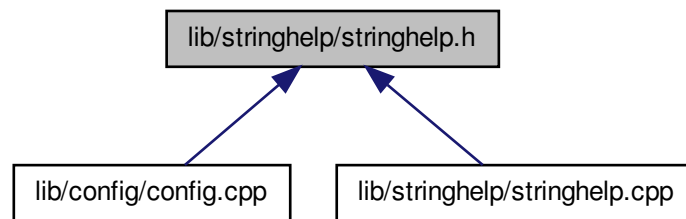
Interface to string helper functions.

```
#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.*

### 6.21.1 Detailed Description

Interface to string helper functions.

#### Author

Paul Griffiths

**Copyright**

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

# Index

- ~Config
  - genleg::Config, 9
- ~DBConnDummy
  - gldb::DBConnDummy, 16
- ~DBConnImp
  - gldb::DBConnImp, 17
- ~DBConnMySQL
  - gldb::DBConnMySQL, 19
- ~Table
  - gldb::Table, 21
- ~TableField
  - gldb::TableField, 24
- ~TableRow
  - gldb::TableRow, 27
- \_XOPEN\_SOURCE
  - config\_getopt.cpp, 34
- add\_cmdline\_option
  - genleg::Config, 10
- append\_field
  - gldb::TableRow, 27, 28
- append\_record
  - gldb::Table, 21
- Config
  - genleg::Config, 9
- config\_getopt.cpp
  - \_XOPEN\_SOURCE, 34
- DBConn
  - gldb::DBConn, 13
- DBConnCouldNotConnect
  - gldb::DBConnCouldNotConnect, 14
- DBConnCouldNotQuery
  - gldb::DBConnCouldNotQuery, 15
- DBConnDummy
  - gldb::DBConnDummy, 16
- DBConnImp
  - gldb::DBConnImp, 17
- DBConnMySQL
  - gldb::DBConnMySQL, 19
- genleg::Config, 9
  - ~Config, 9
  - add\_cmdline\_option, 10
  - Config, 9
  - is\_set, 10
  - m\_opts\_set, 11
  - m\_opts\_supp, 11
  - populate\_from\_cmdline, 10
  - populate\_from\_file, 11
- genleg::ConfigBadConfigFile, 11
- genleg::ConfigBadOption, 11
- genleg::ConfigCouldNotOpenFile, 12
- genleg::ConfigOptionNotSet, 12
- get\_headers
  - gldb::Table, 21
- gldb::DBConn, 12
  - DBConn, 13
  - m\_imp, 14
  - operator=, 13
  - select, 13
- gldb::DBConnCouldNotConnect, 14
  - DBConnCouldNotConnect, 14
- gldb::DBConnCouldNotQuery, 14
  - DBConnCouldNotQuery, 15
- gldb::DBConnDummy, 15
  - ~DBConnDummy, 16
  - DBConnDummy, 16
  - operator=, 16
  - select, 16
- gldb::DBConnImp, 17
  - ~DBConnImp, 17
  - DBConnImp, 17
  - select, 17
- gldb::DBConnMySQL, 18
  - ~DBConnMySQL, 19
  - DBConnMySQL, 19
  - m\_conn, 20
  - operator=, 19
  - select, 19
- gldb::Table, 20
  - ~Table, 21
  - append\_record, 21
  - get\_headers, 21
  - m\_headers, 22
  - m\_records, 22
  - num\_fields, 21
  - num\_records, 22
  - Table, 21
- gldb::TableField, 22
  - ~TableField, 24
  - length, 24
  - m\_data, 26
  - operator std::string, 24
  - operator<<, 26
  - operator+=, 24, 25
  - operator=, 25
  - TableField, 24

- gldb::TableRow, 26
  - ~TableRow, 27
  - append\_field, 27, 28
  - m\_fields, 29
  - print, 28
  - size, 28
  - TableRow, 27
- is\_set
  - genleg::Config, 10
- length
  - gldb::TableField, 24
- lib/config/config.cpp, 31
- lib/config/config.h, 32
- lib/config/config\_getopt.cpp, 33
- lib/database/data\_structures.h, 34
- lib/database/database.h, 36
- lib/database/dbconn.cpp, 37
- lib/database/dbconn.h, 39
- lib/database/dbconnimp.h, 40
- lib/database/table.cpp, 43
- lib/database/table.h, 44
- lib/database/tablefield.cpp, 45
- lib/database/tablefield.h, 46
- lib/database/ablerow.cpp, 48
- lib/database/ablerow.h, 49
- lib/database\_imp/database\_imp.h, 50
- lib/database\_imp/dummy/dbconn\_dummy\_imp.cpp, 52
- lib/database\_imp/dummy/dbconn\_dummy\_imp.h, 54
- lib/database\_imp/mysql/dbconn\_mysql\_imp.cpp, 55
- lib/database\_imp/mysql/dbconn\_mysql\_imp.h, 56
- lib/stringhelp/stringhelp.cpp, 58
- lib/stringhelp/stringhelp.h, 58
- m\_conn
  - gldb::DBConnMySQL, 20
- m\_data
  - gldb::TableField, 26
- m\_fields
  - gldb::TableRow, 29
- m\_headers
  - gldb::Table, 22
- m\_imp
  - gldb::DBConn, 14
- m\_opts\_set
  - genleg::Config, 11
- m\_opts\_supp
  - genleg::Config, 11
- m\_records
  - gldb::Table, 22
- num\_fields
  - gldb::Table, 21
- num\_records
  - gldb::Table, 22
- operator std::string
  - gldb::TableField, 24
- operator<<
  - gldb::TableField, 26
- operator+=
  - gldb::TableField, 24, 25
- operator=
  - gldb::DBConn, 13
  - gldb::DBConnDummy, 16
  - gldb::DBConnMySQL, 19
  - gldb::TableField, 25
- populate\_from\_cmdline
  - genleg::Config, 10
- populate\_from\_file
  - genleg::Config, 11
- print
  - gldb::TableRow, 28
- select
  - gldb::DBConn, 13
  - gldb::DBConnDummy, 16
  - gldb::DBConnImp, 17
  - gldb::DBConnMySQL, 19
- size
  - gldb::TableRow, 28
- Table
  - gldb::Table, 21
- TableField
  - gldb::TableField, 24
- TableRow
  - gldb::TableRow, 27