general_ledger

Generated by Doxygen 1.8.1.2

Thu Jun 12 2014 22:56:33

Contents

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

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Class Index

3.1 Class Hierarchy

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

Chapter 4

Class Index

4.1 Class List

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

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Chapter 6

Namespace Documentation

6.1 genleg Namespace Reference

Classes

- class ConfigOptionNotSet
- class ConfigBadOption
- class ConfigCouldNotOpenFile
- class ConfigBadConfigFile
- class Config

Enumerations

- enum Argument

6.1.1 Enumeration Type Documentation

6.1.1.1 enum genleg::Argument

6.2 gldb Namespace Reference

Classes

- class DBConnCouldNotConnect
- class DBConnCouldNotQuery
- class DBConn
- class DBConnImp
- class Table
- class TableField
- class TableRow
- class DBConnMySQL

Functions

- std::ostream & operator<< (std::ostream &out, const TableField &field)
- DBConnImp * get_connection (const std::string database, const std::string hostname, const std::string username, const std::string password)

6.2.1 Function Documentation

6.2.1.1 DBConnImp * gldb::get_connection (const std::string *database*, const std::string *hostname*, const std::string *username*, const std::string *password*)

6.2.1.2 std::ostream & gldb::operator<< (std::ostream & *out*, const TableField & *field*)

6.3 pgstring Namespace Reference

Functions

- std::string & trim_front (std::string &s)

  Trims leading whitespace from a string.
- std::string & trim_back (std::string &s)

  Trims trailing whitespace from a string.
- std::string & trim (std::string &s)

  Trims leading and trailing whitespace from a string.
- std::vector< std::string > split (const std::string &s, const char delim)

  Splits a delimited string into tokens.

6.3.1 Function Documentation

6.3.1.1 std::vector< std::string > pgstring::split (const std::string & *s*, const char *delim*)

Splits a delimited string into tokens.

Parameters

s	The string to split.
delim	The delimiter character on which to split.

Returns

A vector of tokens.

6.3.1.2 std::string & pgstring::trim (std::string & *s*)

Trims leading and trailing whitespace from a string.

Parameters

s	The string to trim.

Returns

The trimmed string.

6.3.1.3 std::string & pgstring::trim_back (std::string & *s*)

Trims trailing whitespace from a string.

Parameters

s	The string to trim.

Returns

The trimmed string.

6.3.1.4 std::string & pgstring::trim_front (std::string & *s*)

Trims leading whitespace from a string.

Parameters

s	The string to trim.

Returns

The trimmed string.

Chapter 7

Class Documentation

7.1 genleg::Config Class Reference

`#include <config.h>`

Public Member Functions

- Config ()

  Constructor for Config class.
- ∼Config ()

  Destructor for Config class.
- void add_cmdline_option (const std::string option, const enum Argument arg)
- void populate_from_cmdline (const int argc, char ∗const ∗argv)
- void populate_from_file (const std::string filename)
- bool is_set (const std::string option) const
- const std::string & operator[] (const std::string &option) const

7.1.1 Constructor & Destructor Documentation

7.1.1.1 Config::Config ( )

Constructor for Config class.

7.1.1.2 Config::∼Config ( )

Destructor for Config class.

7.1.2 Member Function Documentation

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

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

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

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

7.1.2.5 void Config::populate_from_file (const std::string *filename*)

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

`#include <config.h>`

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

- lib/config/config.h

7.3 genleg::ConfigBadOption Class Reference

`#include <config.h>`

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

- lib/config/config.h

7.4 genleg::ConfigCouldNotOpenFile Class Reference

`#include <config.h>`

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

- lib/config/config.h

7.5 genleg::ConfigOptionNotSet Class Reference

`#include <config.h>`

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

- lib/config/config.h

7.6 gldb::DBConn Class Reference

`#include <dbconn.h>`

Public Member Functions

- DBConn (DBConnImp *imp)
- ~DBConn ()
- Table select (std::string query)
- DBConn (const DBConn &)
- DBConn & operator= (const DBConn &)

7.6.1 Constructor & Destructor Documentation

7.6.1.1 DBConn::DBConn (DBConnImp * *imp*)

7.6.1.2 DBConn::~DBConn ( )

7.6.1.3 gldb::DBConn::DBConn (const DBConn &)

7.6.2 Member Function Documentation

7.6.2.1 DBConn& gldb::DBConn::operator= (const DBConn &)

7.6.2.2 Table DBConn::select (std::string *query*)

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

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

7.7 gldb::DBConnCouldNotConnect Class Reference

`#include <dbconn.h>`

Public Member Functions

- DBConnCouldNotConnect (const std::string &msg)

7.7.1 Constructor & Destructor Documentation

7.7.1.1 gldb::DBConnCouldNotConnect::DBConnCouldNotConnect (const std::string & *msg*) [inline]

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

- lib/database/dbconn.h

7.8 gldb::DBConnCouldNotQuery Class Reference

`#include <dbconn.h>`

Public Member Functions

- DBConnCouldNotQuery (const std::string &msg)

7.8.1 Constructor & Destructor Documentation

7.8.1.1 gldb::DBConnCouldNotQuery::DBConnCouldNotQuery (const std::string & *msg*) [inline]

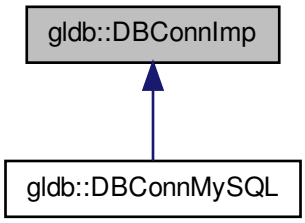
The documentation for this class was generated from the following file:

- lib/database/dbconn.h

7.9 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

7.9.1 Constructor & Destructor Documentation

7.9.1.1 gldb::DBConnImp::DBConnImp ( ) [inline]

7.9.1.2 virtual gldb::DBConnImp::~DBConnImp ( ) [inline], [virtual]

7.9.2 Member Function Documentation

7.9.2.1 virtual Table gldb::DBConnImp::select (std::string *query*) [pure virtual]

Implemented in gldb::DBConnMySQL.

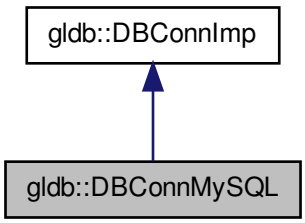
The documentation for this class was generated from the following file:

- lib/database/dbconnimp.h

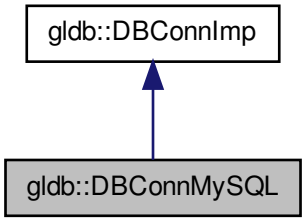
7.10 gldb::DBConnMySQL Class Reference

`#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 for DBConnMySQL_imp class.
- DBConnMySQL (const DBConnMySQL &)
- virtual ~DBConnMySQL ()

  Destructor for DBConnMySQL_imp class.
- DBConnMySQL & operator= (const DBConnMySQL &)
- Table select (std::string query)

7.10.1 Constructor & Destructor Documentation

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

Constructor for DBConnMySQL_imp class.

7.10.1.2 gldb::DBConnMySQL::DBConnMySQL (const DBConnMySQL &)

7.10.1.3 DBConnMySQL::~DBConnMySQL ( ) [virtual]

Destructor for DBConnMySQL_imp class.

7.10.2 Member Function Documentation

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

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

Implements gldb::DBConnImp.

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

7.11 gldb::Table Class Reference

`#include <table.h>`

Public Member Functions

- Table (const TableRow &headers)

  Constructor for Table class.
- ~Table ()

  Destructor for Table class.
- size_t num_fields () const
- size_t num_records () const
- const TableRow & get_headers () const
- const TableRow & operator[] (const size_t idx) const
- void append_record (const TableRow &new_record)

7.11.1 Constructor & Destructor Documentation

7.11.1.1 Table::Table (const TableRow & *headers*)

Constructor for Table class.

7.11.1.2 Table::~Table ( )

Destructor for Table class.

7.11.2 Member Function Documentation

7.11.2.1 void Table::append_record (const TableRow & *new_record*)

7.11.2.2 const TableRow & Table::get_headers ( ) const

7.11.2.3 size_t Table::num_fields ( ) const

7.11.2.4 size_t Table::num_records ( ) const

7.11.2.5 const TableRow & Table::operator[] (const size_t *idx*) const

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

- lib/database/table.h
- lib/database/table.cpp

7.12 gldb::TableField Class Reference

`#include <tablefield.h>`

Public Member Functions

- TableField ()

  Constructor for TableField class.
- TableField (const char *data)
- TableField (const std::string &data)
- ~TableField ()

  Destructor for TableField class.
- size_t length () const
- operator std::string () const
- TableField & operator= (const char *data)
- TableField & operator= (const std::string &data)
- char & operator[] (const size_t idx)
- const char & operator[] (const size_t idx) const
- TableField & operator+= (const char &c)
- TableField & operator+= (const std::string &data)

Friends

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

7.12.1 Constructor & Destructor Documentation

7.12.1.1 TableField::TableField ( )

Constructor for TableField class.

7.12.1.2 TableField::TableField (const char * *data*)

7.12.1.3 TableField::TableField (const std::string & *data*)

7.12.1.4 TableField::~TableField ( )

Destructor for TableField class.

7.12.2 Member Function Documentation

7.12.2.1 size_t TableField::length ( ) const

7.12.2.2 TableField::operator std::string ( ) const

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

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

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

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

7.12.2.7 char & TableField::operator[] (const size_t *idx*)

7.12.2.8 const char & TableField::operator[] (const size_t *idx*) const

7.12.3 Friends And Related Function Documentation

7.12.3.1 std::ostream& operator<< (std::ostream & *out,* const TableField & *field*) [friend]

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

- lib/database/tablefield.h
- lib/database/tablefield.cpp

7.13 gldb::TableRow Class Reference

`#include <tablerow.h>`

Public Member Functions

- TableRow ()

  Constructor for TableRow class.
- TableRow (const size_t size)
- ~TableRow ()

  Destructor for TableRow class.
- size_t size () const
- TableField & operator[] (const size_t idx)
- const TableField & operator[] (const size_t idx) const
- void append_field (const char *new_field)
- void append_field (const std::string &new_field)
- void append_field (const TableField &new_field)
- void print (std::ostream &stream) const

7.13.1 Constructor & Destructor Documentation

7.13.1.1 TableRow::TableRow ( )

Constructor for TableRow class.

7.13.1.2 TableRow::TableRow (const size_t *size*)

7.13.1.3 TableRow::~TableRow ( )

Destructor for TableRow class.

7.13.2 Member Function Documentation

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

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

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

7.13.2.4 **TableField** & TableRow::operator[] (const size_t *idx*)

7.13.2.5 const **TableField** & TableRow::operator[] (const size_t *idx*) const

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

7.13.2.7 size_t TableRow::size ( ) const

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

- lib/database/tablerow.h
- lib/database/tablerow.cpp

Chapter 8

File Documentation

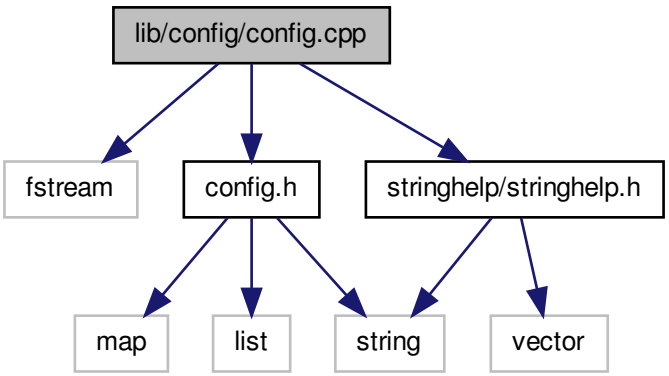
8.1 general_ledger.dox File Reference

8.2 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:



8.2.1 Detailed Description

Implementation of program configurations class.

Author

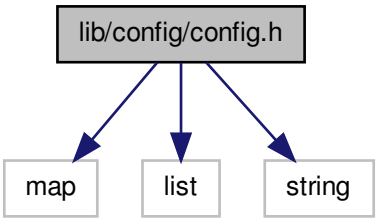
Paul Griffiths

8.3 lib/config/config.h File Reference

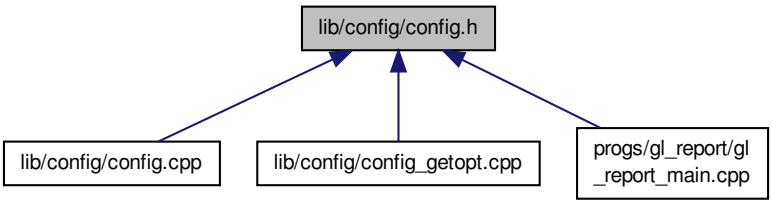
Interface to program configurations class.

```
#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

Namespaces

- namespace genleg

Enumerations

- enum genleg::Argument

8.3.1 Detailed Description

Interface to program configurations class. Interface to program configurations class

Author

Paul Griffiths

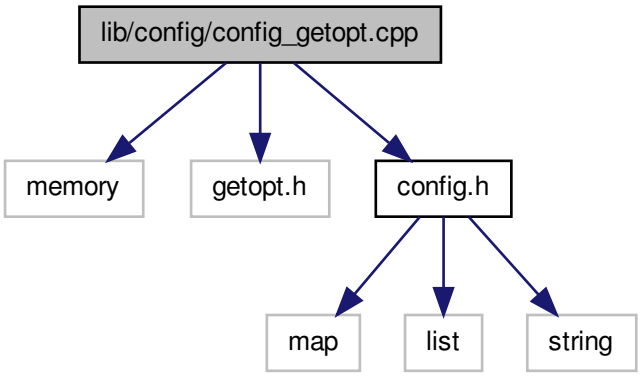
Copyright

8.4 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

8.4.1 Detailed Description

Implementation of command line functionality. Included in separate file to isolate usage of non-standard getopt library.

Author

Paul Griffiths

Copyright

8.4.2 Macro Definition Documentation

8.4.2.1 #define _XOPEN_SOURCE 600

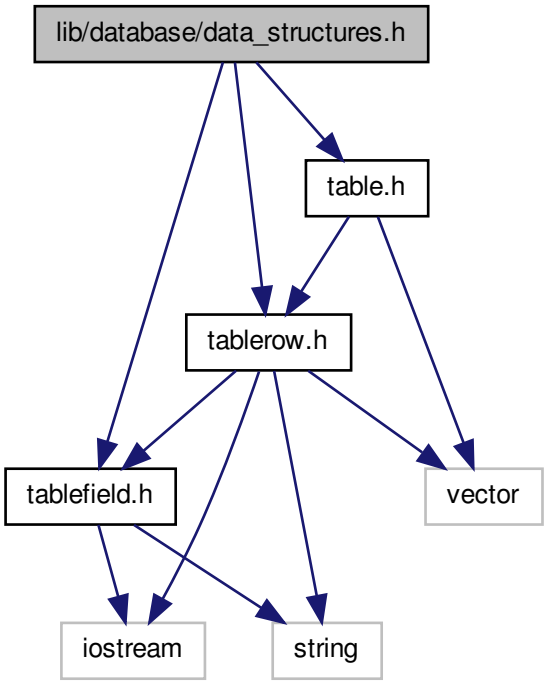
UNIX feature test macro for getopt library

8.5 lib/database/data_structures.h File Reference

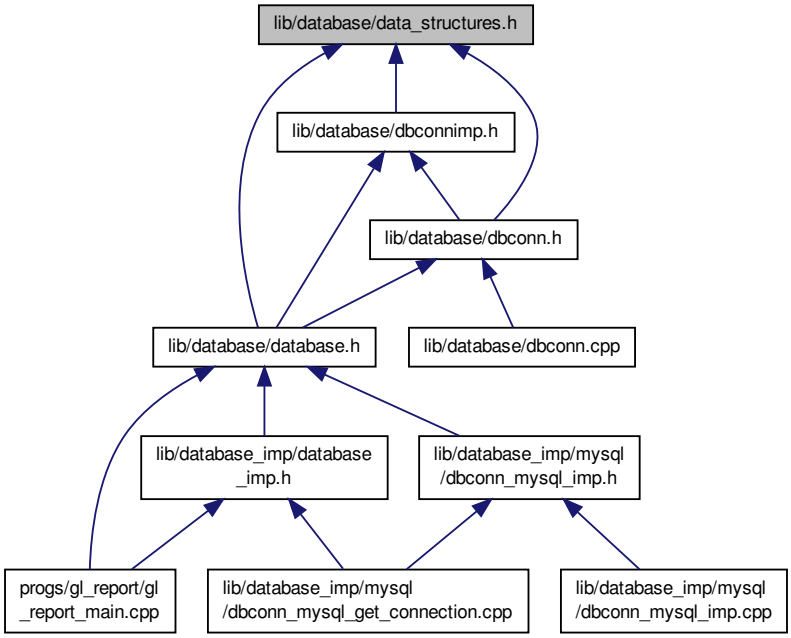
Main interface to database data structures.

```
#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:



8.5.1 Detailed Description

Main interface to database data structures.

Author

Paul Griffiths

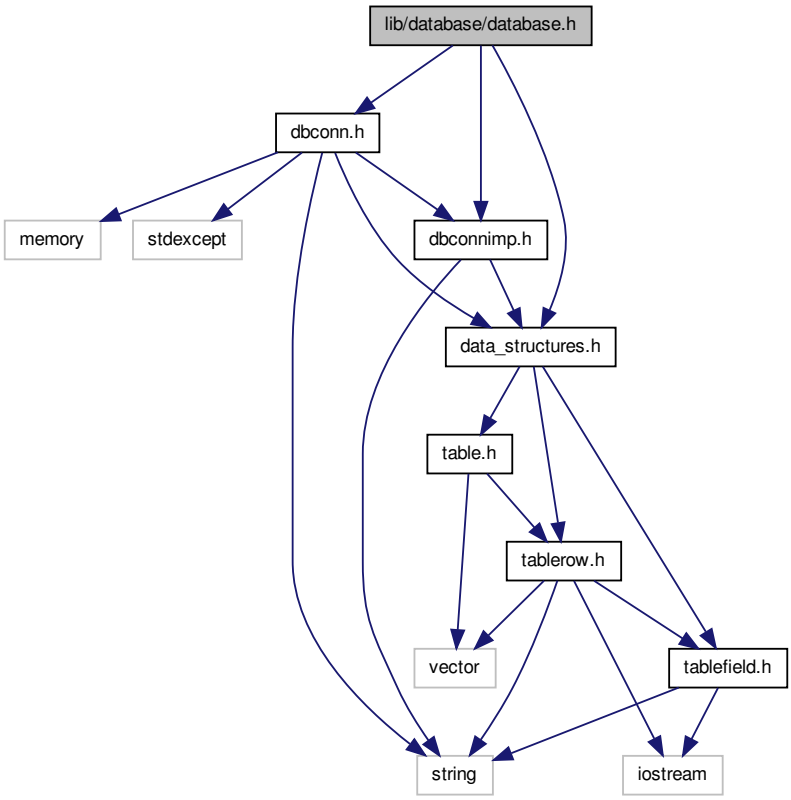
Copyright

8.6 lib/database/database.h File Reference

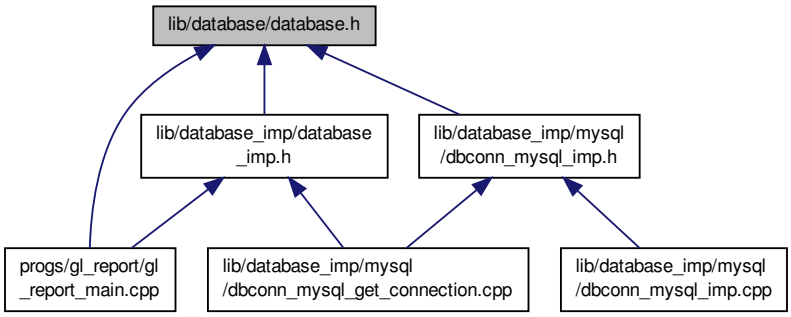
User interface to database functionality.

```
#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:



8.6.1 Detailed Description

User interface to database functionality.

Author

Paul Griffiths

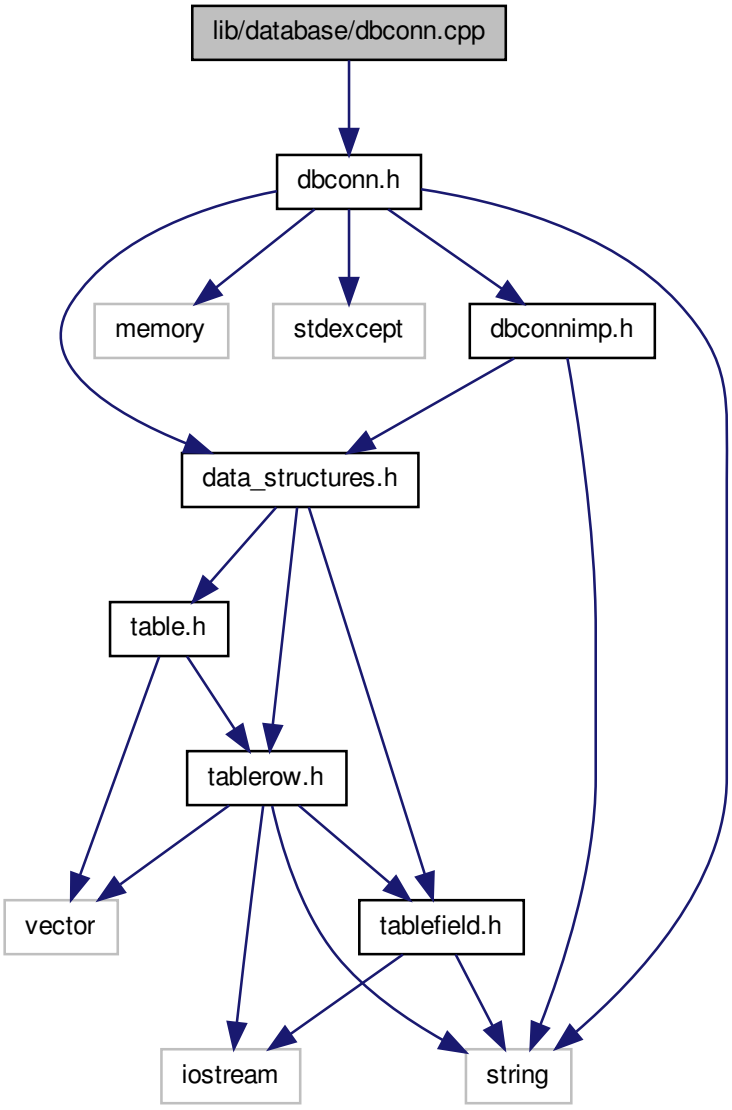
Copyright

8.7 lib/database/dbconn.cpp File Reference

Implementation of database connection abstract base class.

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

Include dependency graph for dbconn.cpp:



8.7.1 Detailed Description

Implementation of database connection abstract base class. Implementation of database connection abstract base class

Author

Paul Griffiths

Copyright

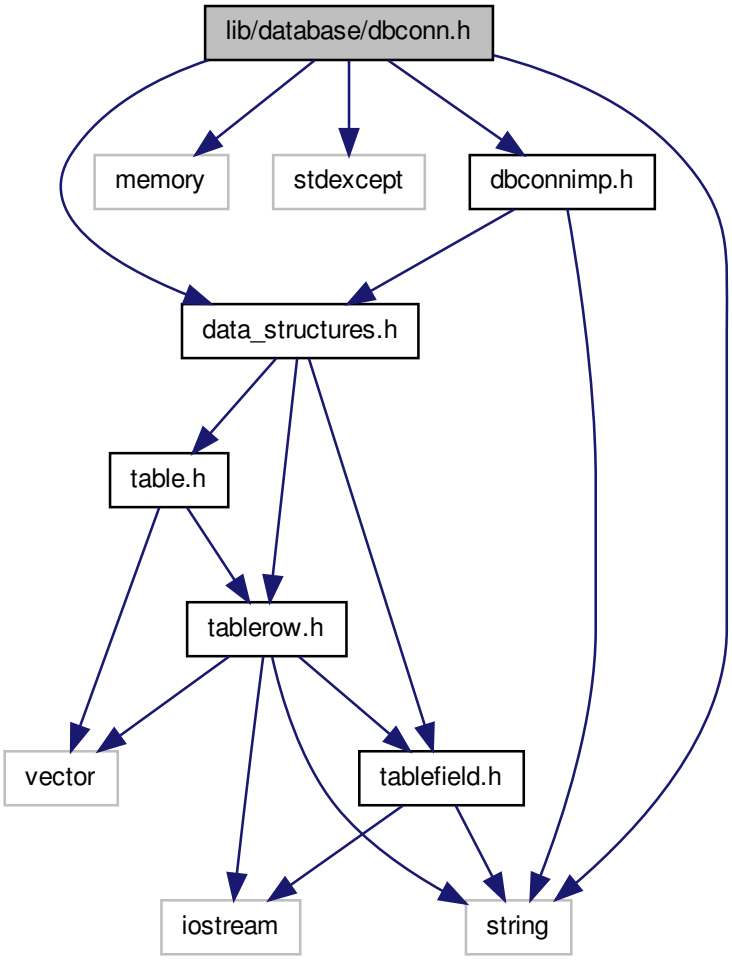
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. http://www.gnu.org/licenses/

8.8 lib/database/dbconn.h File Reference

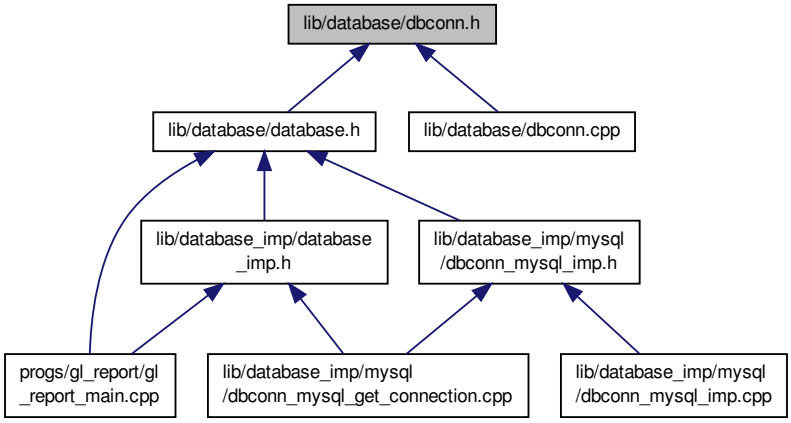
Interface to database connection abstract base class.

```
#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

Namespaces

- namespace gldb

8.8.1 Detailed Description

Interface to database connection abstract base class. Interface to database connection abstract base class

Author

Paul Griffiths

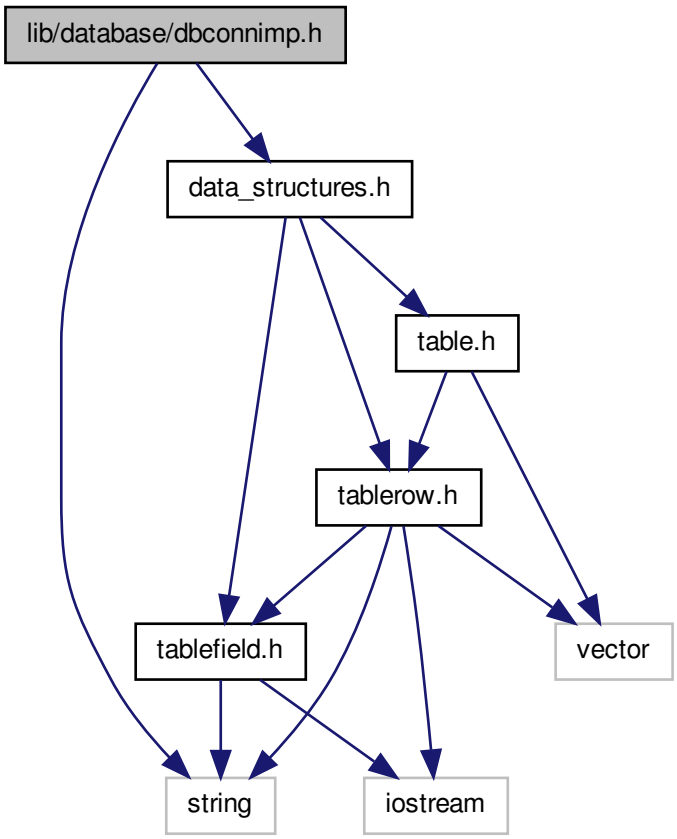
Copyright

8.9 lib/database/dbconnimp.h File Reference

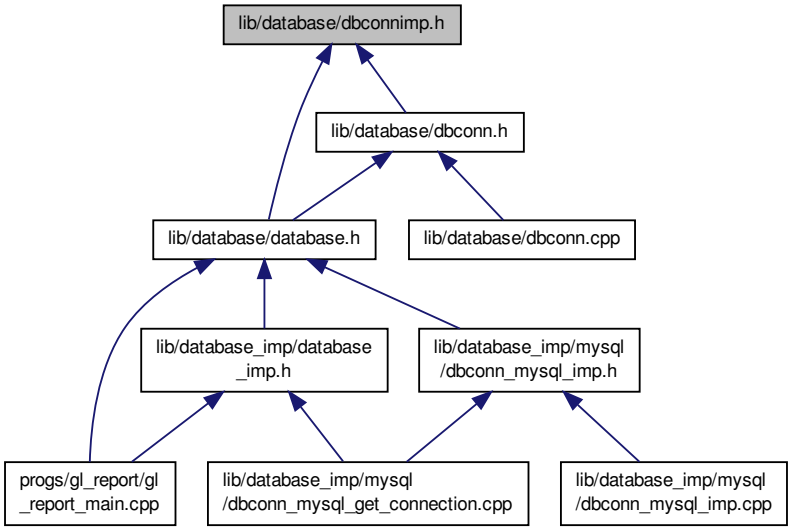
Interface to abstract database implementation base class.

```
#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

Namespaces

- namespace gldb

8.9.1 Detailed Description

Interface to abstract database implementation base class. Interface to abstract database implementation base class

Author

Paul Griffiths

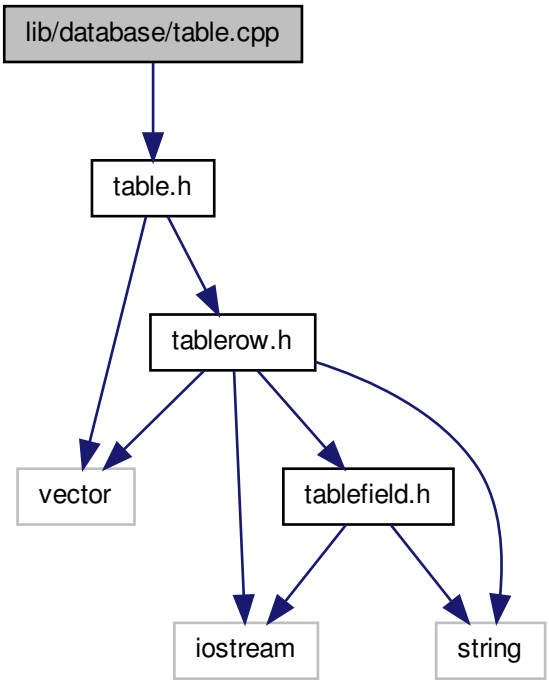
Copyright

8.10 lib/database/table.cpp File Reference

Implementation of table data structure.

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

Include dependency graph for table.cpp:



8.10.1 Detailed Description

Implementation of table data structure. Implementation of table data structure

Author

Paul Griffiths

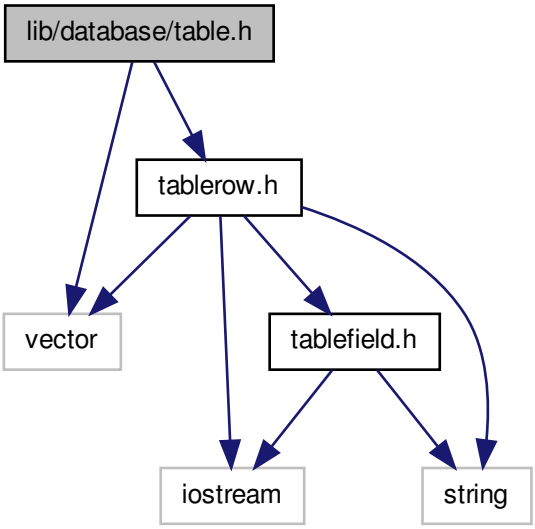
Copyright

8.11 lib/database/table.h File Reference

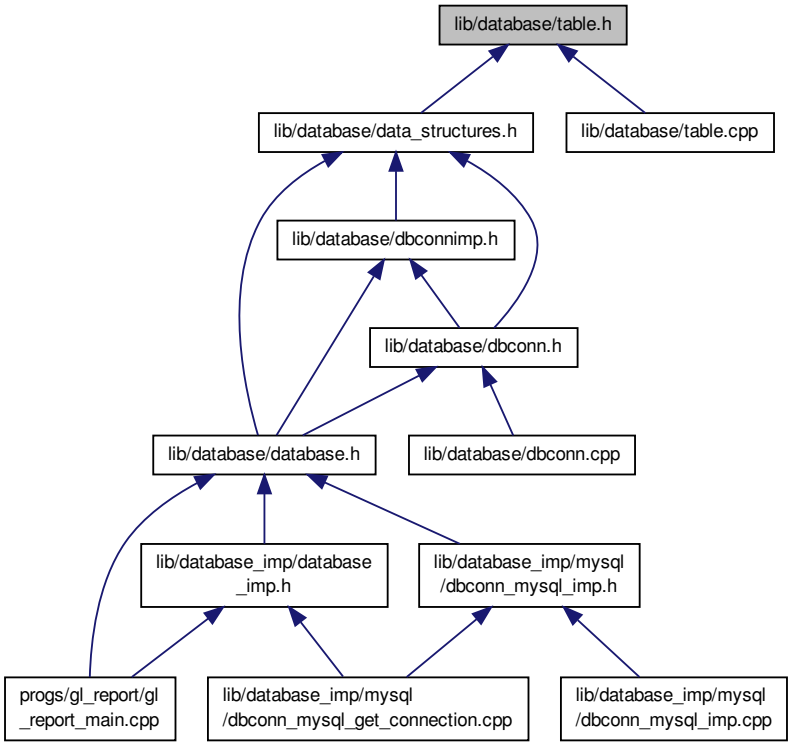
Interface to table data structure.

```
#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

Namespaces

- namespace gldb

8.11.1 Detailed Description

Interface to table data structure. Interface to table data structure

Author

Paul Griffiths

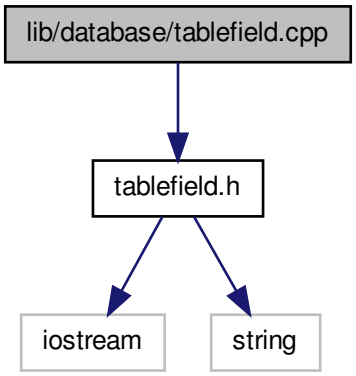
Copyright

8.12 lib/database/tablefield.cpp File Reference

Implementation of table field class.

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

Include dependency graph for tablefield.cpp:



8.12.1 Detailed Description

Implementation of table field class. Implementation of table field class

Author

Paul Griffiths

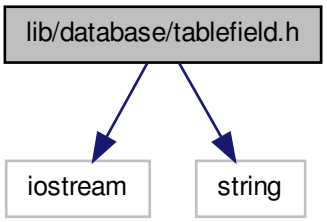
Copyright

8.13 lib/database/tablefield.h File Reference

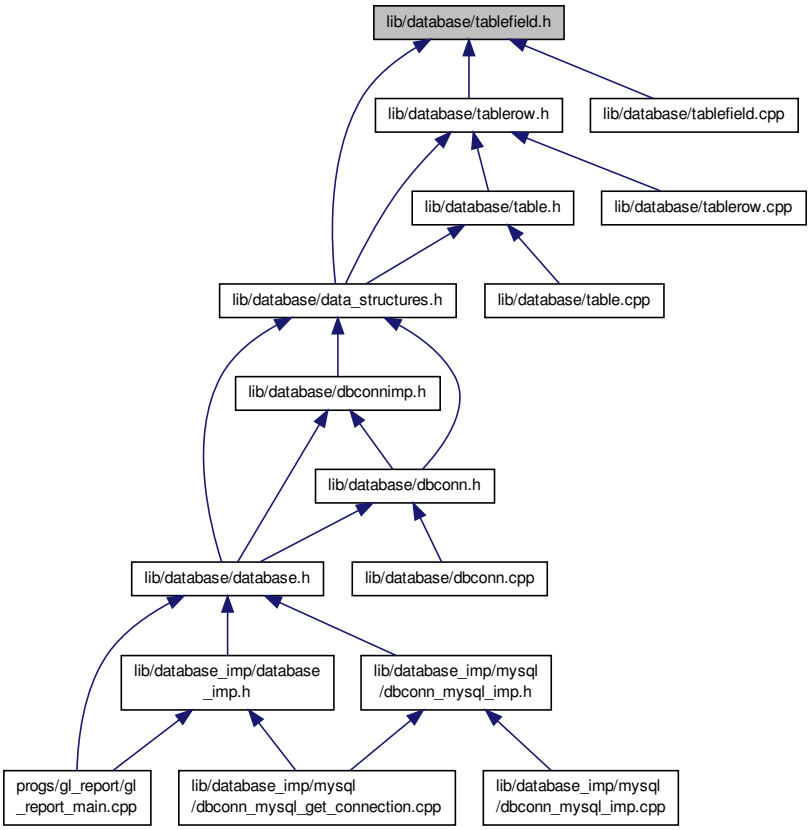
Interface to table field class.

```
#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

Namespaces

- namespace gldb

Functions

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

8.13.1 Detailed Description

Interface to table field class. Interface to table field class

Author

Paul Griffiths

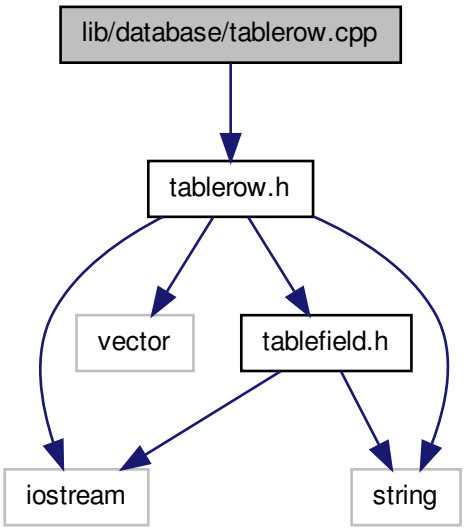
Copyright

8.14 lib/database/tablerow.cpp File Reference

Implementation of table row data structure.

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

Include dependency graph for tablerow.cpp:



8.14.1 Detailed Description

Implementation of table row data structure. Implementation of table row data structure

Author

Paul Griffiths

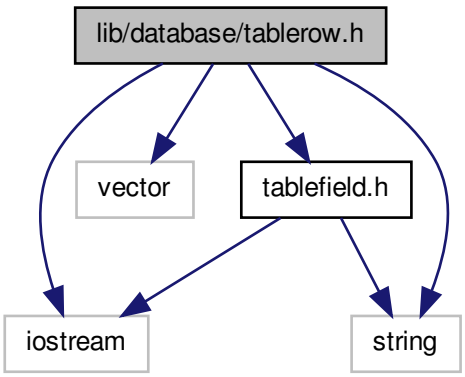
Copyright

8.15 lib/database/tablerow.h File Reference

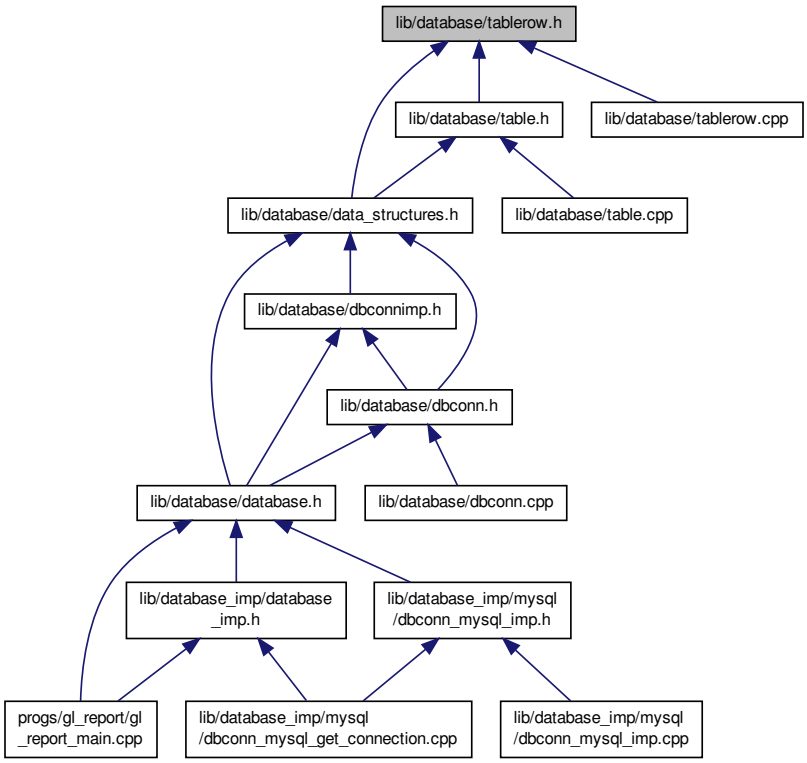
Interface to table row data structure.

```
#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

Namespaces

- namespace gldb

8.15.1 Detailed Description

Interface to table row data structure. Interface to table row data structure

Author

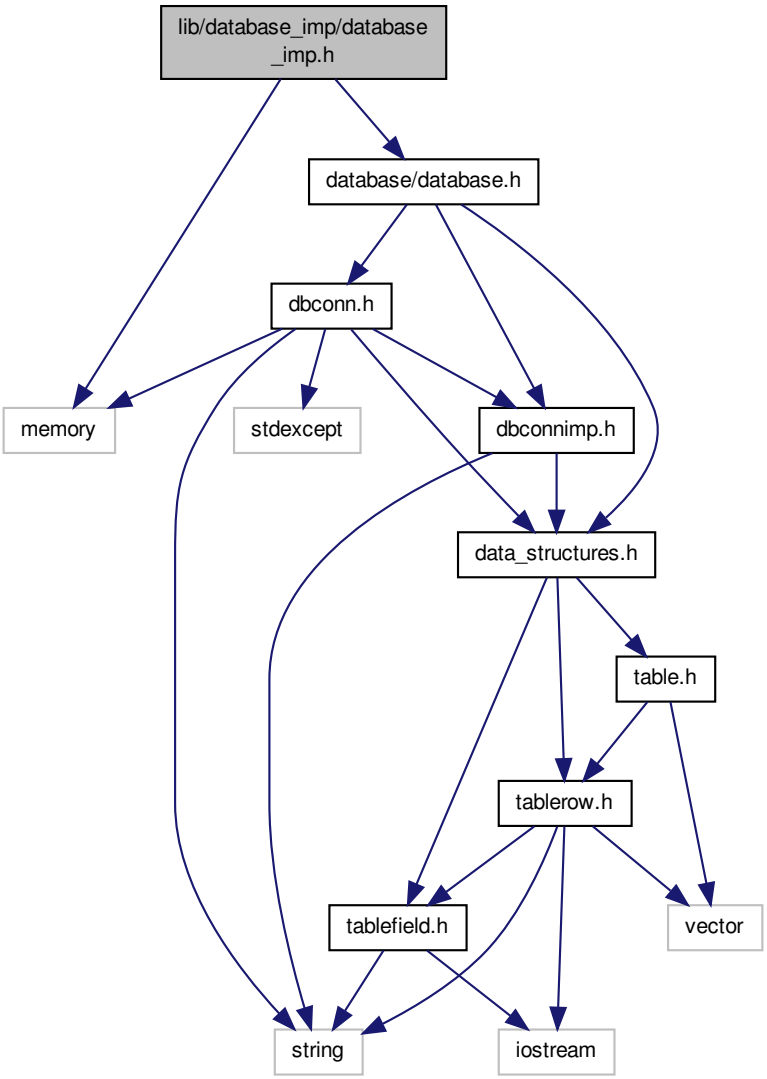
Paul Griffiths

Copyright

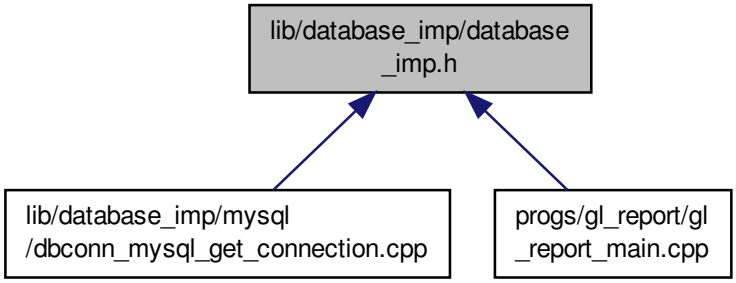
8.16 lib/database_imp/database_imp.h File Reference

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

Include dependency graph for database_imp.h:



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



Namespaces

- namespace gldb

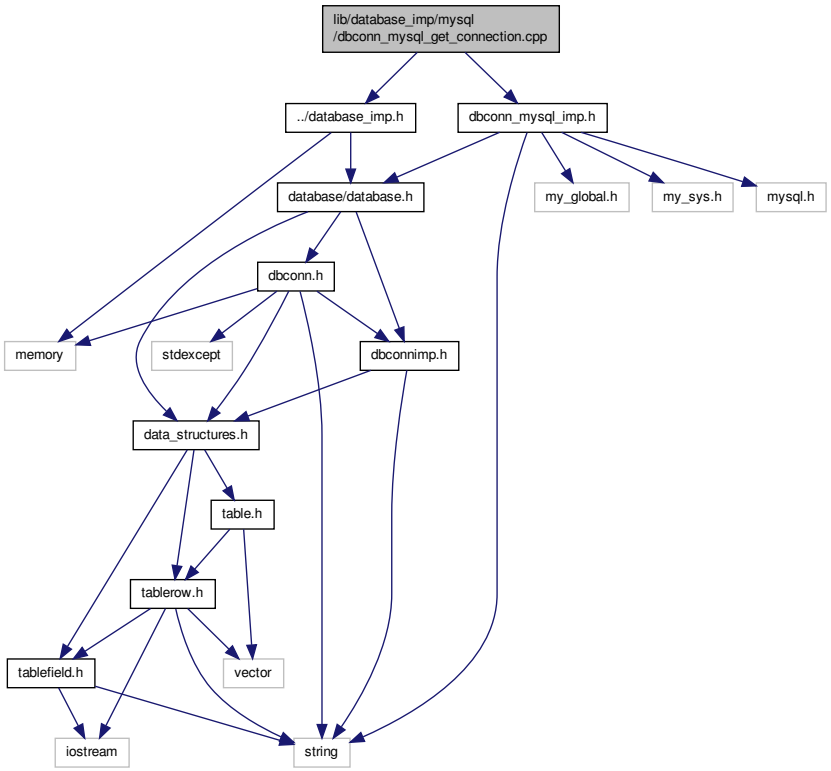
Functions

- DBConnImp * gldb::get_connection (const std::string database, const std::string hostname, const std::string username, const std::string password)

8.17 lib/database_imp/mysql/dbconn_mysql_get_connection.cpp File Reference

```
#include "../database_imp.h"
#include "dbconn_mysql_imp.h"
```

Include dependency graph for dbconn_mysql_get_connection.cpp:

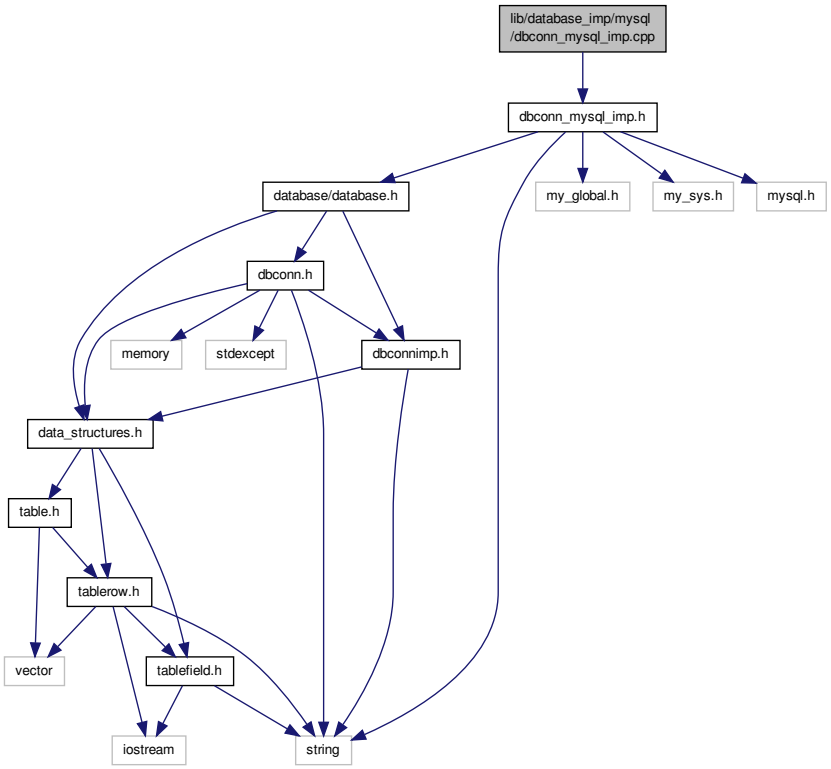


8.18 lib/database_imp/mysql/dbconn_mysql_imp.cpp File Reference

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. Implementation of MySQL database connection implementation class

Author

Paul Griffiths

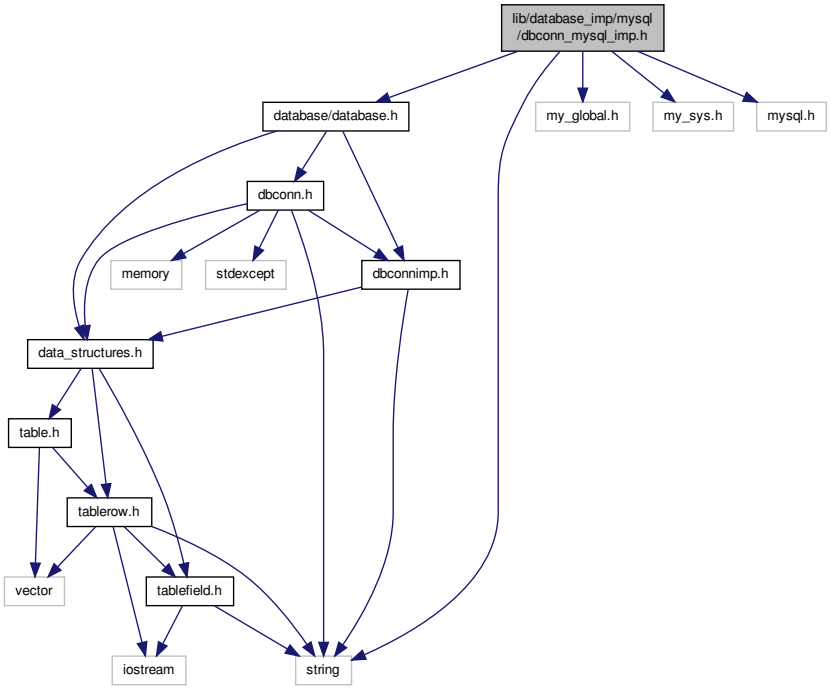
Copyright

8.19 lib/database_imp/mysql/dbconn_mysql_imp.h File Reference

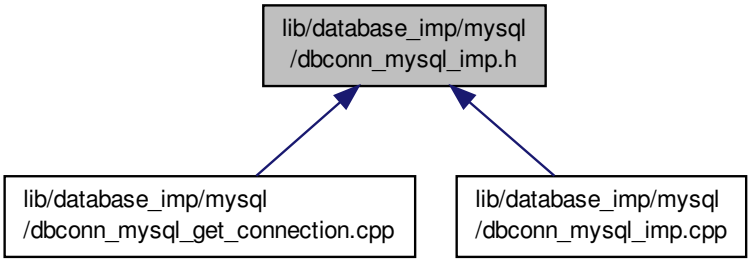
Interface to MySQL database connection implementation class.

```
#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

Namespaces

- namespace gldb

8.19.1 Detailed Description

Interface to MySQL database connection implementation class. Interface to MySQL database connection implementation class

Author

Paul Griffiths

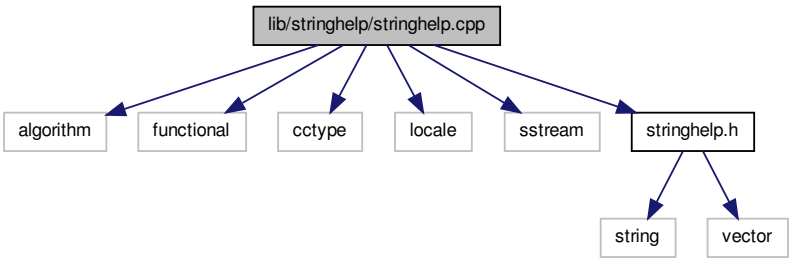
Copyright

8.20 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:



8.20.1 Detailed Description

Implementation of string helper functions.

Author

Paul Griffiths

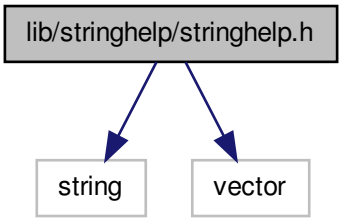
Copyright

8.21 lib/stringhelp/stringhelp.h File Reference

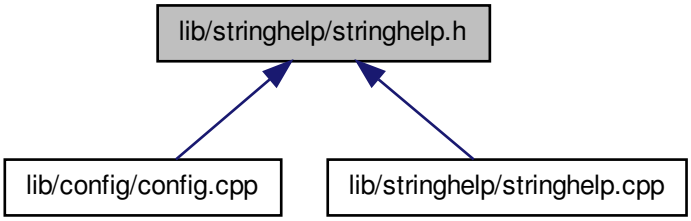
Interface to string helper functions.

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

Include dependency graph for stringhelp.h:



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



Namespaces

- namespace pgstring

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.

8.21.1 Detailed Description

Interface to string helper functions.

Author

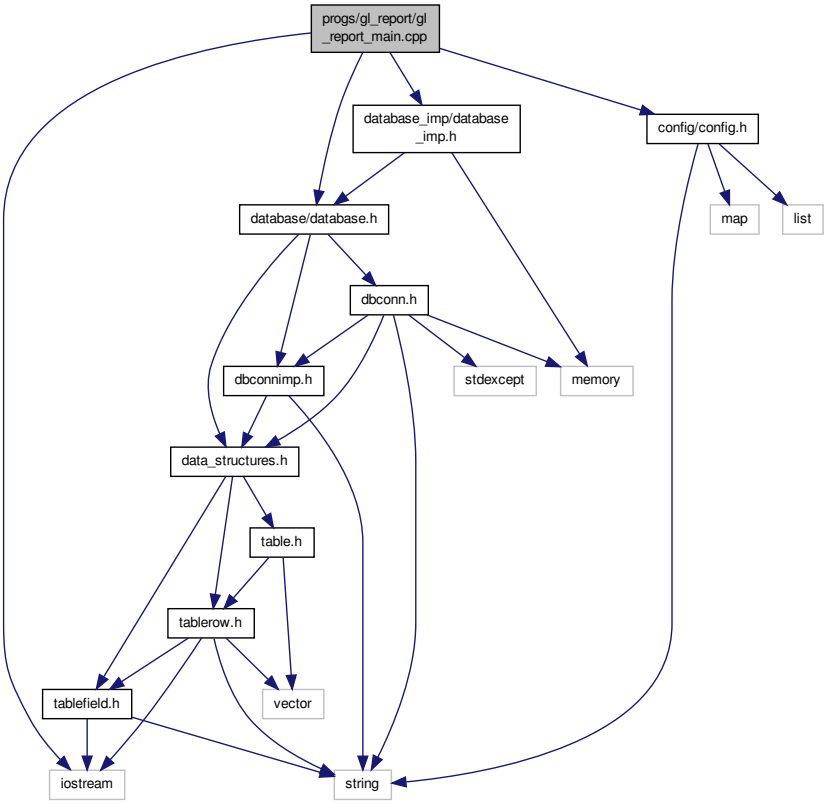
Paul Griffiths

Copyright

8.22 progs/gl_report/gl_report_main.cpp File Reference

```
#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[])
- static void print_usage_message ()

- static void print_version_message ()
- static void print_help_message ()
- static std::string login (void)
- int main (int argc, char ∗argv[])

Variables

- static const char ∗ progname = "gl_report"

8.22.1 Function Documentation

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

8.22.1.2 int main (int *argc,* char ∗ *argv[]*)

8.22.1.3 static void print_help_message ( ) [static]

8.22.1.4 static void print_usage_message ( ) [static]

8.22.1.5 static void print_version_message ( ) [static]

8.22.1.6 static void set_configuration (genleg::Config & *config,* int *argc,* char ∗ *argv[]*) [static]

8.22.2 Variable Documentation

8.22.2.1 const char∗ progname = ”gl_report” [static]

Index