general_ledger

Generated by Doxygen 1.8.1.2

Sun Jun 22 2014 20:08:01

Contents

1 General Ledger. 1

2 Todo List 3

3 Bug List 5

4 Module Index 7

4.1 Modules . . . 7

5 Class Index 9

5.1 Class Hierarchy . . . 9

6 Class Index 11

6.1 Class List . . . 11

7 File Index 13

7.1 File List . . . 13

8 Module Documentation 17

8.1 Program configuration module . . . 17

8.1.1 Detailed Description . . . 17

8.2 Database interaction module . . . 18

8.2.1 Detailed Description . . . 19

8.2.2 Function Documentation . . . 19

8.2.2.1 get_connection . . . 19

8.2.2.2 get_database_type . . . 19

8.2.2.3 get_field_names . . . 19

8.2.2.4 get_row . . . 19

8.3 SQL statements module . . . 21

8.3.1 Detailed Description . . . 21

8.4 General Ledger database module. . . . 22

8.4.1 Detailed Description . . . 22

8.4.2 Function Documentation . . . 23

8.4.2.1 decorated_report_from_table . . . 23

8.4.2.2 decorated_row . . . 23
8.4.2.3 grow_widths . . . 23
8.4.2.4 max_column_widths . . . 23
8.4.2.5 plain_report_from_table . . . 24
8.4.2.6 plain_row . . . 24
8.4.2.7 separator_row . . . 24
8.5 General purpose utilities. . . . 25
8.5.1 Detailed Description . . . 26
8.5.2 Function Documentation . . . 26
8.5.2.1 content_lines . . . 26
8.5.2.2 currency_from_string . . . 26
8.5.2.3 join . . . 26
8.5.2.4 next_content_line . . . 26
8.5.2.5 operator!= . . . 27
8.5.2.6 operator+ . . . 27
8.5.2.7 operator- . . . 27
8.5.2.8 operator< . . . 27
8.5.2.9 operator<= . . . 28
8.5.2.10 operator== . . . 28
8.5.2.11 operator> . . . 28
8.5.2.12 operator>= . . . 29
8.5.2.13 replace . . . 29
8.5.2.14 split . . . 29
8.5.2.15 split . . . 29
8.5.2.16 split_lines . . . 30
8.5.2.17 trim . . . 30
8.5.2.18 trim_back . . . 30
8.5.2.19 trim_front . . . 30
8.6 Database program. . . . 32
8.6.1 Detailed Description . . . 32
8.6.2 Function Documentation . . . 32
8.6.2.1 check_db_parameters . . . 32
8.6.2.2 check_help_and_version . . . 32
8.6.2.3 login . . . 33
8.6.2.4 main . . . 33
8.6.2.5 set_configuration . . . 33
8.7 Reporting program. . . . 34
8.7.1 Detailed Description . . . 34
8.7.2 Function Documentation . . . 34
8.7.2.1 check_db_parameters . . . 34

8.7.2.2 check_help_and_version . . . 34
8.7.2.3 login . . . 35
8.7.2.4 main . . . 35
8.7.2.5 set_configuration . . . 35
8.8 User administration program. . . . 36
8.8.1 Detailed Description . . . 36
8.8.2 Function Documentation . . . 36
8.8.2.1 check_db_parameters . . . 36
8.8.2.2 check_help_and_version . . . 37
8.8.2.3 check_user_password . . . 37
8.8.2.4 enable_user . . . 37
8.8.2.5 get_user . . . 37
8.8.2.6 login . . . 37
8.8.2.7 main . . . 38
8.8.2.8 set_configuration . . . 38
8.8.2.9 set_user_password . . . 38
8.8.2.10 show_user_details . . . 38

9 Class Documentation **39**

9.1 genleg::Config Class Reference . . . 39
9.1.1 Detailed Description . . . 39
9.1.2 Constructor & Destructor Documentation . . . 39
9.1.2.1 Config . . . 39
9.1.2.2 ~Config . . . 40
9.1.3 Member Function Documentation . . . 40
9.1.3.1 add_cmdline_option . . . 40
9.1.3.2 is_set . . . 40
9.1.3.3 operator[] . . . 40
9.1.3.4 populate_from_cmdline . . . 40
9.1.3.5 populate_from_file . . . 41
9.1.4 Member Data Documentation . . . 41
9.1.4.1 m_opts_set . . . 41
9.1.4.2 m_opts_supp . . . 41
9.2 genleg::ConfigBadConfigFile Class Reference . . . 41
9.2.1 Detailed Description . . . 42
9.2.2 Constructor & Destructor Documentation . . . 42
9.2.2.1 ConfigBadConfigFile . . . 42
9.3 genleg::ConfigBadOption Class Reference . . . 43
9.3.1 Detailed Description . . . 43
9.3.2 Constructor & Destructor Documentation . . . 43

9.3.2.1 ConfigBadOption . . . 44
9.4 genleg::ConfigCouldNotOpenFile Class Reference . . . 44
9.4.1 Detailed Description . . . 45
9.4.2 Constructor & Destructor Documentation . . . 45
9.4.2.1 ConfigCouldNotOpenFile . . . 45
9.5 genleg::ConfigException Class Reference . . . 45
9.5.1 Detailed Description . . . 45
9.5.2 Constructor & Destructor Documentation . . . 46
9.5.2.1 ConfigException . . . 46
9.6 genleg::ConfigOptionNotSet Class Reference . . . 46
9.6.1 Detailed Description . . . 47
9.6.2 Constructor & Destructor Documentation . . . 47
9.6.2.1 ConfigOptionNotSet . . . 47
9.7 pgutils::Currency Class Reference . . . 47
9.7.1 Detailed Description . . . 48
9.7.2 Constructor & Destructor Documentation . . . 48
9.7.2.1 Currency . . . 48
9.7.3 Member Function Documentation . . . 48
9.7.3.1 expand . . . 48
9.7.3.2 operator+= . . . 48
9.7.3.3 operator- . . . 48
9.7.3.4 operator-= . . . 49
9.7.4 Friends And Related Function Documentation . . . 49
9.7.4.1 operator+ . . . 49
9.7.4.2 operator< . . . 49
9.7.4.3 operator== . . . 49
9.7.5 Member Data Documentation . . . 50
9.7.5.1 m_frac . . . 50
9.7.5.2 m_int . . . 50
9.8 pgutils::CurrencyException Class Reference . . . 50
9.8.1 Detailed Description . . . 50
9.8.2 Constructor & Destructor Documentation . . . 50
9.8.2.1 CurrencyException . . . 50
9.9 gldb::DBConn Class Reference . . . 50
9.9.1 Detailed Description . . . 51
9.9.2 Constructor & Destructor Documentation . . . 52
9.9.2.1 DBConn . . . 52
9.9.2.2 DBConn . . . 52
9.9.2.3 DBConn . . . 52
9.9.3 Member Function Documentation . . . 52

9.9.3.1 last_auto_increment . . . 52
9.9.3.2 operator= . . . 52
9.9.3.3 operator= . . . 52
9.9.3.4 query . . . 52
9.9.3.5 select . . . 52
9.9.4 Member Data Documentation . . . 53
9.9.4.1 m_imp . . . 53
9.10 gldb::DBConnCouldNotConnect Class Reference . . . 53
9.10.1 Detailed Description . . . 54
9.10.2 Constructor & Destructor Documentation . . . 54
9.10.2.1 DBConnCouldNotConnect . . . 54
9.11 gldb::DBConnCouldNotQuery Class Reference . . . 54
9.11.1 Detailed Description . . . 55
9.11.2 Constructor & Destructor Documentation . . . 55
9.11.2.1 DBConnCouldNotQuery . . . 55
9.12 gldb::DBConnDummy Class Reference . . . 55
9.12.1 Detailed Description . . . 56
9.12.2 Constructor & Destructor Documentation . . . 56
9.12.2.1 DBConnDummy . . . 57
9.12.2.2 DBConnDummy . . . 57
9.12.2.3 ~DBConnDummy . . . 57
9.12.3 Member Function Documentation . . . 57
9.12.3.1 operator= . . . 57
9.12.3.2 query . . . 57
9.12.3.3 select . . . 57
9.13 gldb::DBConnException Class Reference . . . 58
9.13.1 Detailed Description . . . 58
9.13.2 Constructor & Destructor Documentation . . . 58
9.13.2.1 DBConnException . . . 58
9.14 gldb::DBConnImp Class Reference . . . 58
9.14.1 Detailed Description . . . 59
9.14.2 Constructor & Destructor Documentation . . . 59
9.14.2.1 DBConnImp . . . 59
9.14.2.2 ~DBConnImp . . . 59
9.14.3 Member Function Documentation . . . 60
9.14.3.1 last_auto_increment . . . 60
9.14.3.2 query . . . 60
9.14.3.3 select . . . 60
9.15 gldb::DBConnMySQL Class Reference . . . 60
9.15.1 Detailed Description . . . 62

9.15.2 Constructor & Destructor Documentation . . . 62
9.15.2.1 DBConnMySQL . . . 62
9.15.2.2 DBConnMySQL . . . 62
9.15.2.3 DBConnMySQL . . . 62
9.15.2.4 ~DBConnMySQL . . . 62
9.15.3 Member Function Documentation . . . 62
9.15.3.1 last_auto_increment . . . 62
9.15.3.2 operator= . . . 63
9.15.3.3 operator= . . . 63
9.15.3.4 query . . . 63
9.15.3.5 select . . . 63
9.15.4 Member Data Documentation . . . 63
9.15.4.1 m_conn . . . 63
9.15.4.2 mtx . . . 63
9.16 genleg::DBSQLDummy Class Reference . . . 64
9.16.1 Detailed Description . . . 64
9.17 genleg::DBSQLMySQL Class Reference . . . 64
9.17.1 Detailed Description . . . 65
9.18 genleg::DBSQLStatements Class Reference . . . 65
9.18.1 Detailed Description . . . 67
9.18.2 Constructor & Destructor Documentation . . . 67
9.18.2.1 DBSQLStatements . . . 67
9.18.2.2 ~DBSQLStatements . . . 67
9.18.3 Member Function Documentation . . . 67
9.18.3.1 account_by_name . . . 67
9.18.3.2 create_table . . . 67
9.18.3.3 create_view . . . 68
9.18.3.4 currenttb . . . 68
9.18.3.5 currenttb_by_entity . . . 68
9.18.3.6 drop_table . . . 68
9.18.3.7 drop_view . . . 68
9.18.3.8 entity_by_id . . . 69
9.18.3.9 entity_by_name . . . 69
9.18.3.10 get_perms . . . 69
9.18.3.11 grant . . . 69
9.18.3.12 je_by_id . . . 70
9.18.3.13 jelines_by_id . . . 70
9.18.3.14 listusers . . . 70
9.18.3.15 post_je . . . 70
9.18.3.16 post_je_line . . . 71

9.18.3.17 revoke . . . 71
9.18.3.18 standing_data . . . 71
9.18.3.19 update_user . . . 71
9.18.3.20 user_by_id . . . 72
9.18.3.21 user_by_username . . . 72
9.19 genleg::GLAccount Class Reference . . . 72
9.19.1 Detailed Description . . . 73
9.19.2 Constructor & Destructor Documentation . . . 73
9.19.2.1 GLAccount . . . 73
9.19.3 Member Data Documentation . . . 73
9.19.3.1 m_description . . . 73
9.19.3.2 m_enabled . . . 73
9.20 genleg::GLDatabase Class Reference . . . 73
9.20.1 Detailed Description . . . 75
9.20.2 Constructor & Destructor Documentation . . . 75
9.20.2.1 GLDatabase . . . 75
9.20.2.2 ∼GLDatabase . . . 76
9.20.3 Member Function Documentation . . . 76
9.20.3.1 backend . . . 76
9.20.3.2 create_entity . . . 76
9.20.3.3 create_structure . . . 76
9.20.3.4 create_user . . . 76
9.20.3.5 current_trial_balance_report . . . 77
9.20.3.6 destroy_structure . . . 77
9.20.3.7 get_account_by_name . . . 77
9.20.3.8 get_entity_by_id . . . 77
9.20.3.9 get_entity_by_name . . . 78
9.20.3.10 get_je_by_id . . . 78
9.20.3.11 get_standing_data . . . 78
9.20.3.12 get_user_by_id . . . 78
9.20.3.13 get_user_by_username . . . 79
9.20.3.14 grant . . . 79
9.20.3.15 je_report . . . 79
9.20.3.16 list_users_report . . . 79
9.20.3.17 load_sample_data . . . 79
9.20.3.18 post_journal . . . 80
9.20.3.19 report . . . 80
9.20.3.20 revoke . . . 80
9.20.3.21 standing_data_report . . . 80
9.20.3.22 update_user . . . 80

9.20.4 Member Data Documentation . . . 80
9.20.4.1 m_dbc . . . 81
9.20.4.2 m_sql . . . 81
9.20.4.3 m_tables . . . 81
9.20.4.4 m_views . . . 81
9.21 genleg::GLDBException Class Reference . . . 81
9.21.1 Detailed Description . . . 81
9.21.2 Constructor & Destructor Documentation . . . 81
9.21.2.1 GLDBException . . . 81
9.22 genleg::GLDBTransaction Class Reference . . . 82
9.22.1 Detailed Description . . . 82
9.22.2 Constructor & Destructor Documentation . . . 82
9.22.2.1 GLDBTransaction . . . 82
9.22.2.2 ~GLDBTransaction . . . 83
9.22.3 Member Data Documentation . . . 83
9.22.3.1 m_commit . . . 83
9.22.3.2 m_dbc . . . 83
9.23 genleg::GLEntity Class Reference . . . 83
9.23.1 Detailed Description . . . 84
9.23.2 Constructor & Destructor Documentation . . . 84
9.23.2.1 GLEntity . . . 84
9.23.3 Member Data Documentation . . . 84
9.23.3.1 m_aggregate . . . 84
9.23.3.2 m_enabled . . . 84
9.23.3.3 m_name . . . 84
9.23.3.4 m_parent . . . 84
9.23.3.5 m_shortname . . . 84
9.24 genleg::GLJELine Class Reference . . . 85
9.24.1 Detailed Description . . . 85
9.24.2 Constructor & Destructor Documentation . . . 85
9.24.2.1 GLJELine . . . 85
9.24.3 Member Function Documentation . . . 86
9.24.3.1 account . . . 86
9.24.3.2 amount . . . 86
9.24.4 Member Data Documentation . . . 86
9.24.4.1 m_acct . . . 86
9.24.4.2 m_amount . . . 86
9.25 genleg::GLJournal Class Reference . . . 86
9.25.1 Detailed Description . . . 87
9.25.2 Constructor & Destructor Documentation . . . 87

9.25.2.1 GLJournal . . . 87
9.25.3 Member Function Documentation . . . 87
9.25.3.1 begin . . . 87
9.25.3.2 begin . . . 88
9.25.3.3 end . . . 88
9.25.3.4 end . . . 88
9.25.4 Member Data Documentation . . . 88
9.25.4.1 m_entity . . . 88
9.25.4.2 m_id . . . 88
9.25.4.3 m_lines . . . 88
9.25.4.4 m_memo . . . 88
9.25.4.5 m_period . . . 88
9.25.4.6 m_source . . . 89
9.25.4.7 m_user . . . 89
9.25.4.8 m_year . . . 89
9.26 genleg::GLReport Class Reference . . . 89
9.26.1 Detailed Description . . . 89
9.26.2 Constructor & Destructor Documentation . . . 89
9.26.2.1 GLReport . . . 89
9.26.3 Member Data Documentation . . . 89
9.26.3.1 m_headers . . . 89
9.26.3.2 m_report_text . . . 90
9.26.3.3 m_title . . . 90
9.27 genleg::GLStandingData Class Reference . . . 90
9.27.1 Detailed Description . . . 90
9.27.2 Constructor & Destructor Documentation . . . 90
9.27.2.1 GLStandingData . . . 90
9.27.3 Member Data Documentation . . . 90
9.27.3.1 m_num_periods . . . 90
9.27.3.2 m_period . . . 91
9.27.3.3 m_year . . . 91
9.28 genleg::GLUser Class Reference . . . 91
9.28.1 Detailed Description . . . 92
9.28.2 Constructor & Destructor Documentation . . . 92
9.28.2.1 GLUser . . . 92
9.28.2.2 ~GLUser . . . 93
9.28.3 Member Function Documentation . . . 93
9.28.3.1 check_password . . . 93
9.28.3.2 enabled . . . 93
9.28.3.3 firstname . . . 93

9.28.3.4 id . . . 93
9.28.3.5 lastname . . . 93
9.28.3.6 pass_hash . . . 94
9.28.3.7 pass_salt . . . 94
9.28.3.8 permissions . . . 94
9.28.3.9 set_enabled . . . 94
9.28.3.10 set_firstname . . . 94
9.28.3.11 set_lastname . . . 94
9.28.3.12 set_password . . . 94
9.28.3.13 set_username . . . 95
9.28.3.14 username . . . 95
9.28.4 Member Data Documentation . . . 95
9.28.4.1 m_enabled . . . 95
9.28.4.2 m_firstname . . . 95
9.28.4.3 m_id . . . 95
9.28.4.4 m_lastname . . . 95
9.28.4.5 m_pass_hash . . . 95
9.28.4.6 m_pass_salt . . . 95
9.28.4.7 m_perms . . . 95
9.28.4.8 m_username . . . 96
9.29 gldb::MySQLResult Class Reference . . . 96
9.29.1 Detailed Description . . . 96
9.29.2 Constructor & Destructor Documentation . . . 96
9.29.2.1 MySQLResult . . . 96
9.29.2.2 ~MySQLResult . . . 97
9.29.2.3 MySQLResult . . . 97
9.29.2.4 MySQLResult . . . 97
9.29.3 Member Function Documentation . . . 97
9.29.3.1 num_fields . . . 97
9.29.3.2 operator= . . . 97
9.29.3.3 operator= . . . 97
9.29.3.4 result . . . 97
9.29.4 Member Data Documentation . . . 97
9.29.4.1 m_num_fields . . . 97
9.29.4.2 m_result . . . 97
9.30 gldb::Table Class Reference . . . 98
9.30.1 Detailed Description . . . 99
9.30.2 Constructor & Destructor Documentation . . . 99
9.30.2.1 Table . . . 99
9.30.2.2 Table . . . 99

9.30.2.3 Table . . . 99
9.30.2.4 Table . . . 100
9.30.2.5 ~Table . . . 100
9.30.3 Member Function Documentation . . . 100
9.30.3.1 append_record . . . 100
9.30.3.2 append_record . . . 100
9.30.3.3 begin . . . 100
9.30.3.4 begin . . . 100
9.30.3.5 create_from_file . . . 101
9.30.3.6 end . . . 101
9.30.3.7 end . . . 101
9.30.3.8 get_field . . . 101
9.30.3.9 get_headers . . . 102
9.30.3.10 insert_query . . . 102
9.30.3.11 num_fields . . . 102
9.30.3.12 num_records . . . 102
9.30.3.13 operator= . . . 102
9.30.3.14 operator= . . . 102
9.30.3.15 operator[] . . . 103
9.30.3.16 set_quoted . . . 103
9.30.3.17 set_quoted . . . 103
9.30.4 Member Data Documentation . . . 103
9.30.4.1 m_headers . . . 103
9.30.4.2 m_quoted . . . 103
9.30.4.3 m_records . . . 103
9.31 gldb::TableBadInputFile Class Reference . . . 104
9.31.1 Detailed Description . . . 104
9.31.2 Constructor & Destructor Documentation . . . 104
9.31.2.1 TableBadInputFile . . . 105
9.32 gldb::TableCouldNotOpenInputFile Class Reference . . . 105
9.32.1 Detailed Description . . . 106
9.32.2 Constructor & Destructor Documentation . . . 106
9.32.2.1 TableCouldNotOpenInputFile . . . 106
9.33 gldb::TableException Class Reference . . . 106
9.33.1 Detailed Description . . . 107
9.33.2 Constructor & Destructor Documentation . . . 107
9.33.2.1 TableException . . . 107
9.34 gldb::TableField Class Reference . . . 107
9.34.1 Detailed Description . . . 108
9.34.2 Constructor & Destructor Documentation . . . 108

9.34.2.1 TableField . . . 108
9.34.2.2 TableField . . . 108
9.34.2.3 TableField . . . 109
9.34.2.4 TableField . . . 109
9.34.2.5 TableField . . . 109
9.34.2.6 ∼TableField . . . 109
9.34.3 Member Function Documentation . . . 109
9.34.3.1 length . . . 109
9.34.3.2 operator std::string . . . 109
9.34.3.3 operator+= . . . 109
9.34.3.4 operator+= . . . 110
9.34.3.5 operator= . . . 110
9.34.3.6 operator= . . . 110
9.34.3.7 operator= . . . 110
9.34.3.8 operator= . . . 111
9.34.3.9 operator= . . . 111
9.34.3.10 operator[] . . . 111
9.34.3.11 operator[] . . . 111
9.34.4 Friends And Related Function Documentation . . . 112
9.34.4.1 operator<< . . . 112
9.34.5 Member Data Documentation . . . 112
9.34.5.1 m_data . . . 112
9.35 gldb::TableMismatchedRecordLength Class Reference . . . 112
9.35.1 Detailed Description . . . 113
9.35.2 Constructor & Destructor Documentation . . . 113
9.35.2.1 TableMismatchedRecordLength . . . 113
9.36 gldb::TableNoSuchField Class Reference . . . 113
9.36.1 Detailed Description . . . 114
9.36.2 Constructor & Destructor Documentation . . . 114
9.36.2.1 TableNoSuchField . . . 114
9.37 gldb::TableNoSuchRecord Class Reference . . . 115
9.37.1 Detailed Description . . . 115
9.37.2 Constructor & Destructor Documentation . . . 115
9.37.2.1 TableNoSuchRecord . . . 116
9.38 gldb::TableRow Class Reference . . . 116
9.38.1 Detailed Description . . . 117
9.38.2 Constructor & Destructor Documentation . . . 117
9.38.2.1 TableRow . . . 117
9.38.2.2 TableRow . . . 117
9.38.2.3 TableRow . . . 117

9.38.2.4 TableRow . . . 117
9.38.2.5 TableRow . . . 118
9.38.2.6 TableRow . . . 118
9.38.2.7 TableRow . . . 118
9.38.2.8 ~TableRow . . . 118
9.38.3 Member Function Documentation . . . 118
9.38.3.1 append_field . . . 118
9.38.3.2 append_field . . . 118
9.38.3.3 append_field . . . 119
9.38.3.4 append_field . . . 119
9.38.3.5 append_field . . . 119
9.38.3.6 begin . . . 119
9.38.3.7 begin . . . 119
9.38.3.8 end . . . 119
9.38.3.9 end . . . 119
9.38.3.10 operator= . . . 120
9.38.3.11 operator= . . . 120
9.38.3.12 operator[] . . . 120
9.38.3.13 operator[] . . . 120
9.38.3.14 print . . . 121
9.38.3.15 record_string . . . 121
9.38.3.16 record_string . . . 121
9.38.3.17 size . . . 121
9.38.4 Member Data Documentation . . . 121
9.38.4.1 m_fields . . . 121

10 File Documentation **123**

10.1 lib/config/config.cpp File Reference . . . 123
10.1.1 Detailed Description . . . 123
10.2 lib/config/config.h File Reference . . . 124
10.2.1 Detailed Description . . . 125
10.3 lib/config/config_getopt.cpp File Reference . . . 125
10.3.1 Detailed Description . . . 125
10.3.2 Macro Definition Documentation . . . 126
10.3.2.1 _XOPEN_SOURCE . . . 126
10.4 lib/database/data_structures.h File Reference . . . 126
10.4.1 Detailed Description . . . 127
10.5 lib/database/database.h File Reference . . . 127
10.5.1 Detailed Description . . . 129
10.6 lib/database/dbconn.cpp File Reference . . . 129

10.6.1 Detailed Description . . . 129
10.7 lib/database/dbconn.h File Reference . . . 130
10.7.1 Detailed Description . . . 131
10.8 lib/database/dbconnimp.h File Reference . . . 131
10.8.1 Detailed Description . . . 133
10.9 lib/database/table.cpp File Reference . . . 133
10.9.1 Detailed Description . . . 133
10.10 lib/database/table.h File Reference . . . 134
10.10.1 Detailed Description . . . 135
10.11 lib/database/tablefield.cpp File Reference . . . 136
10.11.1 Detailed Description . . . 136
10.12 lib/database/tablefield.h File Reference . . . 136
10.12.1 Detailed Description . . . 138
10.13 lib/database/tablerow.cpp File Reference . . . 138
10.13.1 Detailed Description . . . 138
10.14 lib/database/tablerow.h File Reference . . . 139
10.14.1 Detailed Description . . . 140
10.15 lib/database_imp/database_imp.h File Reference . . . 140
10.15.1 Detailed Description . . . 142
10.16 lib/database_imp/dummy/dbconn_dummy_imp.cpp File Reference . . . 142
10.16.1 Detailed Description . . . 143
10.17 lib/database_imp/dummy/dbconn_dummy_imp.h File Reference . . . 143
10.17.1 Detailed Description . . . 145
10.18 lib/database_imp/mysql/dbconn_mysql_functions.cpp File Reference . . . 145
10.18.1 Detailed Description . . . 146
10.19 lib/database_imp/mysql/dbconn_mysql_imp.cpp File Reference . . . 146
10.19.1 Detailed Description . . . 147
10.20 lib/database_imp/mysql/dbconn_mysql_imp.h File Reference . . . 148
10.20.1 Detailed Description . . . 149
10.21 lib/database_imp/mysql/dbconn_mysql_result.cpp File Reference . . . 149
10.21.1 Detailed Description . . . 150
10.22 lib/database_imp/mysql/dbconn_mysql_result.h File Reference . . . 150
10.22.1 Detailed Description . . . 151
10.23 lib/dbsql/dbsql.h File Reference . . . 151
10.23.1 Detailed Description . . . 152
10.24 lib/dbsql/dbsql_dummy.h File Reference . . . 152
10.24.1 Detailed Description . . . 153
10.25 lib/dbsql/dbsql_functions.h File Reference . . . 154
10.25.1 Detailed Description . . . 155
10.26 lib/dbsql/dbsql_implementations.h File Reference . . . 155

10.26.1 Detailed Description . . . 156
10.27 lib/dbsql/dbsql_mysql.h File Reference . . . 157
10.27.1 Detailed Description . . . 158
10.28 lib/dbsql/dbsqlstatements.cpp File Reference . . . 158
10.28.1 Detailed Description . . . 158
10.29 lib/dbsql/dbsqlstatements.h File Reference . . . 159
10.29.1 Detailed Description . . . 160
10.30 lib/gldb/glaccount.cpp File Reference . . . 160
10.30.1 Detailed Description . . . 160
10.31 lib/gldb/glaccount.h File Reference . . . 160
10.31.1 Detailed Description . . . 161
10.32 lib/gldb/gldatabase.cpp File Reference . . . 162
10.32.1 Detailed Description . . . 162
10.32.2 Function Documentation . . . 162
10.32.2.1 boolstring_to_bool . . . 163
10.33 lib/gldb/gldatabase.h File Reference . . . 163
10.33.1 Detailed Description . . . 164
10.34 lib/gldb/gldb.h File Reference . . . 164
10.34.1 Detailed Description . . . 165
10.35 lib/gldb/glentity.cpp File Reference . . . 165
10.35.1 Detailed Description . . . 166
10.36 lib/gldb/glentity.h File Reference . . . 166
10.36.1 Detailed Description . . . 167
10.37 lib/gldb/glexception.h File Reference . . . 167
10.37.1 Detailed Description . . . 168
10.38 lib/gldb/gljournal.cpp File Reference . . . 169
10.38.1 Detailed Description . . . 169
10.39 lib/gldb/gljournal.h File Reference . . . 169
10.39.1 Detailed Description . . . 171
10.40 lib/gldb/glreport.cpp File Reference . . . 171
10.40.1 Detailed Description . . . 172
10.41 lib/gldb/glreport.h File Reference . . . 172
10.41.1 Detailed Description . . . 174
10.42 lib/gldb/glstanding.cpp File Reference . . . 174
10.42.1 Detailed Description . . . 175
10.43 lib/gldb/glstanding.h File Reference . . . 175
10.43.1 Detailed Description . . . 176
10.44 lib/gldb/gluser.cpp File Reference . . . 176
10.44.1 Detailed Description . . . 176
10.45 lib/gldb/gluser.h File Reference . . . 177

10.45.1 Detailed Description . . . 177
10.46 lib/gldb/gluser_pass.cpp File Reference . . . 178
10.46.1 Detailed Description . . . 178
10.46.2 Macro Definition Documentation . . . 179
10.46.2.1 _XOPEN_SOURCE . . . 179
10.46.3 Function Documentation . . . 179
10.46.3.1 generate_salt . . . 179
10.47 lib/pgutils/currency.cpp File Reference . . . 179
10.47.1 Detailed Description . . . 179
10.48 lib/pgutils/currency.h File Reference . . . 180
10.48.1 Detailed Description . . . 181
10.49 lib/pgutils/pgutils.h File Reference . . . 182
10.49.1 Detailed Description . . . 182
10.50 lib/pgutils/stringhelp.cpp File Reference . . . 183
10.50.1 Detailed Description . . . 183
10.51 lib/pgutils/stringhelp.h File Reference . . . 183
10.51.1 Detailed Description . . . 185
10.52 progs/gl_db/gl_db_main.cpp File Reference . . . 185
10.52.1 Detailed Description . . . 186
10.53 progs/gl_report/gl_report_main.cpp File Reference . . . 186
10.53.1 Detailed Description . . . 187
10.54 progs/gl_user/gl_user_main.cpp File Reference . . . 188
10.54.1 Detailed Description . . . 189

Chapter 1

General Ledger.

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

Chapter 2

Todo List

File gluser_pass.cpp

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

Chapter 3

Bug List

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

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

Chapter 4

Module Index

4.1 Modules

Here is a list of all modules:

Program configuration module 17
Database interaction module 18
SQL statements module 21
General Ledger database module. 22
General purpose utilities. 25
Database program. 32
Reporting program. 34
User administration program. 36

Chapter 5

Class Index

5.1 Class Hierarchy

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

genleg::Config . . . 39
genleg::ConfigException . . . 45
- genleg::ConfigBadConfigFile . . . 41
- genleg::ConfigBadOption . . . 43
- genleg::ConfigCouldNotOpenFile . . . 44
- genleg::ConfigOptionNotSet . . . 46

pgutils::Currency . . . 47
pgutils::CurrencyException . . . 50
gldb::DBConn . . . 50
gldb::DBConnException . . . 58
- gldb::DBConnCouldNotConnect . . . 53
- gldb::DBConnCouldNotQuery . . . 54

gldb::DBConnImp . . . 58
- gldb::DBConnDummy . . . 55
- gldb::DBConnMySQL . . . 60

genleg::DBSQLStatements . . . 65
- genleg::DBSQLDummy . . . 64
- genleg::DBSQLMySQL . . . 64

genleg::GLAccount . . . 72
genleg::GLDatabase . . . 73
genleg::GLDBException . . . 81
genleg::GLDBTransaction . . . 82
genleg::GLEntity . . . 83
genleg::GLJELine . . . 85
genleg::GLJournal . . . 86
genleg::GLReport . . . 89
genleg::GLStandingData . . . 90
genleg::GLUser . . . 91
gldb::MySQLResult . . . 96
gldb::Table . . . 98
gldb::TableException . . . 106
- gldb::TableBadInputFile . . . 104
- gldb::TableCouldNotOpenInputFile . . . 105
- gldb::TableMismatchedRecordLength . . . 112
- gldb::TableNoSuchField . . . 113
- gldb::TableNoSuchRecord . . . 115

gldb::TableField 107
gldb::TableRow 116

Chapter 6

Class Index

6.1 Class List

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

genleg::Config
: Configuration options class . . . 39

genleg::ConfigBadConfigFile
: Exception class for badly formed configuration file . . . 41

genleg::ConfigBadOption
: Exception class for bad provided option . . . 43

genleg::ConfigCouldNotOpenFile
: Exception class for when conf file cannot be opened . . . 44

genleg::ConfigException
: Configuration module exception base class . . . 45

genleg::ConfigOptionNotSet
: Exception class for option not set . . . 46

pgutils::Currency
: Currency amount class . . . 47

pgutils::CurrencyException
: Base Currency exception class . . . 50

gldb::DBConn
: Database connection class . . . 50

gldb::DBConnCouldNotConnect
: Could not connect to database exception class . . . 53

gldb::DBConnCouldNotQuery
: Could not execute database query exception class . . . 54

gldb::DBConnDummy
: Dummy database implementation class . . . 55

gldb::DBConnException
: Base database connection exception class . . . 58

gldb::DBConnImp
: Abstract database implementation base class . . . 58

gldb::DBConnMySQL
: MySQL database implementation class . . . 60

genleg::DBSQLDummy
: Dummy SQL statements class . . . 64

genleg::DBSQLMySQL
: MySQL SQL statements class . . . 64

genleg::DBSQLStatements
: SQL statements class . . . 65

genleg::GLAccount
: Nominal account class . . . 72

genleg::GLDatabase
General ledger database class . . . 73
genleg::GLDBException
Base general ledger database exceptionc class . . . 81
genleg::GLDBTransaction
Database transaction RAII class . . . 82
genleg::GLEntity
General ledger entity class . . . 83
genleg::GLJELine
Journal entry line class . . . 85
genleg::GLJournal
Journal entry class . . . 86
genleg::GLReport
General ledger report class . . . 89
genleg::GLStandingData
General ledger standing data class . . . 90
genleg::GLUser
General ledger user class . . . 91
gldb::MySQLResult
MySQL result structure class . . . 96
gldb::Table
Database table class . . . 98
gldb::TableBadInputFile
Could not connect to database exception class . . . 104
gldb::TableCouldNotOpenInputFile
Could not connect to database exception class . . . 105
gldb::TableException
Base database connection exception class . . . 106
gldb::TableField
Database table field class . . . 107
gldb::TableMismatchedRecordLength
Mismatched record length exception class . . . 112
gldb::TableNoSuchField
No such field exception class . . . 113
gldb::TableNoSuchRecord
No such record exception class . . . 115
gldb::TableRow
Database table row class . . . 116

Chapter 7

File Index

7.1 File List

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

lib/config/config.cpp
Implementation of program configurations class . . . 123
lib/config/config.h
Interface to program configurations class . . . 124
lib/config/config_getopt.cpp
Implementation of command line functionality . . . 125
lib/database/data_structures.h
Main interface to database data structures . . . 126
lib/database/database.h
User interface to database functionality . . . 127
lib/database/dbconn.cpp
Implementation of database connection class . . . 129
lib/database/dbconn.h
Interface to database connection base class . . . 130
lib/database/dbconnimp.h
Interface to abstract database implementation base class . . . 131
lib/database/table.cpp
Implementation of database table data structure . . . 133
lib/database/table.h
Interface to database table data structure . . . 134
lib/database/tablefield.cpp
Implementation of database table field class . . . 136
lib/database/tablefield.h
Interface to database table field class . . . 136
lib/database/tablerow.cpp
Implementation of database table row data structure . . . 138
lib/database/tablerow.h
Interface to database table row data structure . . . 139
lib/database_imp/database_imp.h
Interface to database implementation factory function . . . 140
lib/database_imp/dummy/dbconn_dummy_imp.cpp
Implementation of Dummy database connection implementation class . . . 142
lib/database_imp/dummy/dbconn_dummy_imp.h
Interface to dummy database connection implementation class . . . 143
lib/database_imp/mysql/dbconn_mysql_functions.cpp
Implementation of MySQL implementation factory function . . . 145
lib/database_imp/mysql/dbconn_mysql_imp.cpp
Implementation of MySQL database connection implementation class . . . 146

lib/database_imp/mysql/dbconn_mysql_imp.h
Interface to MySQL database connection implementation class . . . 148
lib/database_imp/mysql/dbconn_mysql_result.cpp
Implementation of MySQL result structure resource handle class . . . 149
lib/database_imp/mysql/dbconn_mysql_result.h
Interface to MySQL result structure resource handle class . . . 150
lib/dbsql/dbsql.h
User interface to DBSQL module . . . 151
lib/dbsql/dbsql_dummy.h
Interface to dummy SQL statement class . . . 152
lib/dbsql/dbsql_functions.h
Interface to SQL module standalone functions . . . 154
lib/dbsql/dbsql_implementations.h
Aggregation header for DBSqlStatements implementations . . . 155
lib/dbsql/dbsql_mysql.h
Interface to MySQL SQL statement class . . . 157
lib/dbsql/dbsqlstatements.cpp
Implementation of SQL statement class . . . 158
lib/dbsql/dbsqlstatements.h
Implementation of SQL module standalone functions . . . 159
lib/gldb/glaccount.cpp
Implementation of nominal account class . . . 160
lib/gldb/glaccount.h
Interface to nominal account class . . . 160
lib/gldb/gldatabase.cpp
Implementation of General Ledger database class . . . 162
lib/gldb/gldatabase.h
Interface to General Ledger database class . . . 163
lib/gldb/gldb.h
User interface to General Ledger database module . . . 164
lib/gldb/glentity.cpp
Implementation of general ledger entity class . . . 165
lib/gldb/glentity.h
Interface to general ledger entity class . . . 166
lib/gldb/glexception.h
Interface to General Ledger base exception class . . . 167
lib/gldb/gljournal.cpp
Implementation of journal entry classes . . . 169
lib/gldb/gljournal.h
Interface to journal entry classes . . . 169
lib/gldb/glreport.cpp
Implementation of report class . . . 171
lib/gldb/glreport.h
Interface to report class . . . 172
lib/gldb/glstanding.cpp
Implementation of general ledger standing data class . . . 174
lib/gldb/glstanding.h
Interface to general ledger standing data class . . . 175
lib/gldb/gluser.cpp
Implementation of user class . . . 176
lib/gldb/gluser.h
Interface to user class . . . 177
lib/gldb/gluser_pass.cpp
Implementation of password functions for user class . . . 178
lib/pgutils/currency.cpp
Implementation of currency amount class . . . 179
lib/pgutils/currency.h
Interface to currency amount class . . . 180

lib/pgutils/pgutils.h
Aggregate interface to general utility functions 182
lib/pgutils/stringhelp.cpp
Implementation of string helper functions 183
lib/pgutils/stringhelp.h
Interface to string helper functions 183
progs/gl_db/gl_db_main.cpp
Main functionality for gl_db program 185
progs/gl_report/gl_report_main.cpp
Main functionality for gl_report program 186
progs/gl_user/gl_user_main.cpp
Main functionality for gl_user program 188

Chapter 8

Module Documentation

8.1 Program configuration module

Classes

- class genleg::ConfigException

  Configuration module exception base class.
- class genleg::ConfigOptionNotSet

  Exception class for option not set.
- class genleg::ConfigBadOption

  Exception class for bad provided option.
- class genleg::ConfigCouldNotOpenFile

  Exception class for when conf file cannot be opened.
- class genleg::ConfigBadConfigFile

  Exception class for badly formed configuration file.
- class genleg::Config

  Configuration options class.

Enumerations

- enum genleg::Argument

  Enumeration class for option argument specifications.

8.1.1 Detailed Description

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

8.2 Database interaction module

Classes

- class gldb::DBConnException

  Base database connection exception class.
- class gldb::DBConnCouldNotConnect

  Could not connect to database exception class.
- class gldb::DBConnCouldNotQuery

  Could not execute database query exception class.
- class gldb::DBConn

  Database connection class.
- class gldb::DBConnImp

  Abstract database implementation base class.
- class gldb::TableException

  Base database connection exception class.
- class gldb::TableNoSuchField

  No such field exception class.
- class gldb::TableNoSuchRecord

  No such record exception class.
- class gldb::TableMismatchedRecordLength

  Mismatched record length exception class.
- class gldb::TableBadInputFile

  Could not connect to database exception class.
- class gldb::TableCouldNotOpenInputFile

  Could not connect to database exception class.
- class gldb::Table

  Database table class.
- class gldb::TableField

  Database table field class.
- class gldb::TableRow

  Database table row class.
- class gldb::DBConnDummy

  Dummy database implementation class.
- class gldb::DBConnMySQL

  MySQL database implementation class.
- class gldb::MySQLResult

  MySQL result structure class.

Functions

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

  Creates and returns a pointer to a database implementation.
- std::string gldb::get_database_type ()

  Returns the name of the compiled-in database type.
- static TableRow get_field_names (MySQLResult &result)

  Gets field names from a MySQL result structure.
- static TableRow get_row (MySQLResult &result, MYSQL_ROW row)

  Creates a TableRow from a MySQL result row.

8.2.1 Detailed Description

Module for interacting with the database.

8.2.2 Function Documentation

8.2.2.1 DBConnImp * 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.

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 only 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

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

Returns

A pointer to the database implementation.

8.2.2.2 std::string gldb::get_database_type ( )

Returns the name of the compiled-in database type.

Returns

The name of the compiled-in database type.

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

Gets field names from a MySQL result structure.

Parameters

result	The MySQL result structure.

Returns

A TableRow containing the field names.

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

Creates a TableRow from a MySQL result row.

Parameters

result	The MySQL result structure.
row	The MySQL row structure.

Returns

A TableRow containing the row data.

8.3 SQL statements module

Classes

- class genleg::DBSQLDummy

  Dummy SQL statements class.
- class genleg::DBSQLMySQL

  MySQL SQL statements class.
- class genleg::DBSQLStatements

  SQL statements class.

8.3.1 Detailed Description

Module for producing SQL statements used by program.

8.4 General Ledger database module.

Classes

- class genleg::GLAccount

  Nominal account class.
- class genleg::GLDatabase

  General ledger database class.
- class genleg::GLDBTransaction

  Database transaction RAII class.
- class genleg::GLEntity

  General ledger entity class.
- class genleg::GLDBException

  Base general ledger database exceptionc class.
- class genleg::GLJELine

  Journal entry line class.
- class genleg::GLJournal

  Journal entry class.
- class genleg::GLReport

  General ledger report class.
- class genleg::GLStandingData

  General ledger standing data class.
- class genleg::GLUser

  General ledger user class.
- class pgutils::CurrencyException

  Base Currency exception class.

Functions

- static std::vector< size_t > max_column_widths (const gldb::Table &table)

  Calculates the maximum required column widths for a table.
- static void grow_widths (std::vector< size_t > &widths, const TableRow &row)

  Increments a vector of required column widths.
- static std::string separator_row (const std::vector< size_t > &widths)

  Returns a decorated separator row for a table.
- static std::string plain_row (const TableRow &row, const std::vector< size_t > &widths)

  Returns a row for a plain report.
- static std::string decorated_row (const TableRow &row, const std::vector< size_t > &widths)

  Returns a row for a decorated report.
- std::string genleg::plain_report_from_table (const gldb::Table &table)

  Creates a plain report from a table.
- std::string genleg::decorated_report_from_table (const gldb::Table &table)

  Creates a decorated report from a table.

8.4.1 Detailed Description

Module for interacting with the general ledger database model.

8.4.2 Function Documentation

8.4.2.1 std::string genleg::decorated_report_from_table (const gldb::Table & *table*)

Creates a decorated report from a table.

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

Parameters

table	The table from which to create the report.

Returns

A string containing the report.

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

Returns a row for a decorated report.

Parameters

row	The row for which to create the report row.
widths	A vector of required widths.

Returns

A string containing the decorated row.

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

Increments a vector of required column widths.

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

Parameters

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

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

Calculates the maximum required column widths for a table.

Parameters

table	The table.

Returns

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

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

Creates a plain report from a table.

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

Parameters

table	The table from which to create the report.

Returns

A string containing the report.

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

Returns a row for a plain report.

Parameters

row	The row for which to create the report row.
widths	A vector of required widths.

Returns

A string containing the plain row.

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

Returns a decorated separator row for a table.

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

Parameters

widths	A vector of required widths.

Returns

A string containing the separator row.

8.5 General purpose utilities.

Classes

- class pgutils::Currency

  Currency amount class.

Functions

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

  Currency addition operator.
- Currency pgutils::operator- (Currency lhs, const Currency &rhs)

  Currency subtraction operator.
- bool pgutils::operator== (const Currency &lhs, const Currency &rhs)

  Currency equality comparison operator.
- bool pgutils::operator!= (const Currency &lhs, const Currency &rhs)

  Currency inequality comparison operator.
- bool pgutils::operator< (const Currency &lhs, const Currency &rhs)

  Currency less than comparison operator.
- bool pgutils::operator> (const Currency &lhs, const Currency &rhs)

  Currency greater than comparison operator.
- bool pgutils::operator<= (const Currency &lhs, const Currency &rhs)

  Currency less than or equal to comparison operator.
- bool pgutils::operator>= (const Currency &lhs, const Currency &rhs)

  Currency greater than or equal to comparison operator.
- Currency pgutils::currency_from_string (const std::string &s)

  Creates a currency amount from a string representation.
- std::string & pgutils::trim_front (std::string &s)

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

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

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

  Splits a delimited string into tokens.
- std::vector< std::string > & pgutils::split (std::vector< std::string > &vec, const std::string &s, const char delim)

  Splits a delimited string into tokens.
- bool pgutils::next_content_line (std::istream &ifs, std::string &s)

  Gets the next content line from a stream.
- std::vector< std::string > & pgutils::content_lines (std::vector< std::string > &vec, std::istream &ifs)

  Populates a vector of content lines from a stream.
- std::vector< std::vector < std::string > > & pgutils::split_lines (std::vector< std::vector< std::string >> &vec, std::istream &ifs, const char delim)

  Populates a vector of vectors of fields from a stream.
- std::string & pgutils::join (const std::vector< std::string > &vec, std::string &s, const char delim)

  Joins a vector of strings into a delimited line.
- bool pgutils::replace (std::string &str, const std::string &from, const std::string &to)

  Replaces a substring with another string.

8.5.1 Detailed Description

General purpose utility classes and functions.

8.5.2 Function Documentation

8.5.2.1 std::vector< std::string > & pgutils::content_lines (std::vector< std::string > & *vec,* std::istream & *ifs*)

Populates a vector of content lines from a stream.

Parameters

vec	The vector to populate.
ifs	The input stream.

Returns

A reference to `vec`.

8.5.2.2 Currency pgutils::currency_from_string (const std::string & *s*)

Creates a currency amount from a string representation.

Parameters

s	The string representation.

Returns

The currency representation.

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

Joins a vector of strings into a delimited line.

The function is the opposite of `split`.

Parameters

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

Returns

A reference to `s`.

8.5.2.4 bool pgutils::next_content_line (std::istream & *ifs,* std::string & *s*)

Gets the next content line from a stream.

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

Parameters

ifs	The input stream.
s	The string in which to store the line.

Returns

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

8.5.2.5 bool pgutils::operator!= (const Currency & *lhs*, const Currency & *rhs*)

Currency inequality comparison operator.

Parameters

lhs	Left hand side.
rhs	Right hand side.

Return values

true	If the two sides are not equal.
false	If the two sides are equal.

8.5.2.6 Currency pgutils::operator+ (Currency *lhs*, const Currency & *rhs*)

Currency addition operator.

Parameters

lhs	Left hand side.
rhs	Right hand side.

Returns

The sum of the two sides.

8.5.2.7 Currency pgutils::operator- (Currency *lhs*, const Currency & *rhs*)

Currency subtraction operator.

Parameters

lhs	Left hand side.
rhs	Right hand side.

Returns

The difference between the two sides.

8.5.2.8 bool pgutils::operator< (const Currency & *lhs*, const Currency & *rhs*)

Currency less than comparison operator.

Parameters

lhs	Left hand side.
rhs	Right hand side.

Return values

true	If the lhs is less than the rhs.
false	If the lhs is not less than the rhs.

8.5.2.9 bool pgutils::operator<= (const Currency & *lhs*, const Currency & *rhs*)

Currency less than or equal to comparison operator.

Parameters

lhs	Left hand side.
rhs	Right hand side.

Return values

true	If the lhs is less than or equal to the rhs.
false	If the lhs is not less than or equal to the rhs.

8.5.2.10 bool pgutils::operator== (const Currency & *lhs*, const Currency & *rhs*)

Currency equality comparison operator.

Parameters

lhs	Left hand side.
rhs	Right hand side.

Return values

true	If the two sides are equal.
false	If the two sides are not equal.

8.5.2.11 bool pgutils::operator> (const Currency & *lhs*, const Currency & *rhs*)

Currency greater than comparison operator.

Parameters

lhs	Left hand side.
rhs	Right hand side.

Return values

true	If the lhs is greater than the rhs.
false	If the lhs is not greater than the rhs.

8.5.2.12 bool pgutils::operator>= (const Currency & *lhs*, const Currency & *rhs*)

Currency greater than or equal to comparison operator.

Parameters

lhs	Left hand side.
rhs	Right hand side.

Return values

true	If the lhs is greater than or equal to the rhs.
false	If the lhs is not greater than or equal to the rhs.

8.5.2.13 bool pgutils::replace (std::string & *str*, const std::string & *from*, const std::string & *to*)

Replaces a substring with another string.

Parameters

str	The string containing the substring to replace.
from	The substring to replace.
to	The string with which to replace the substring.

Returns

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

8.5.2.14 std::vector< std::string > pgutils::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.

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

Splits a delimited string into tokens.

Parameters

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

Returns

A reference to `vec`.

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

Populates a vector of vectors of fields from a stream.

Parameters

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

Returns

A reference to `vec`.

8.5.2.17 std::string & pgutils::trim (std::string & *s*)

Trims leading and trailing whitespace from a string.

Parameters

s	The string to trim.

Returns

The trimmed string.

8.5.2.18 std::string & pgutils::trim_back (std::string & *s*)

Trims trailing whitespace from a string.

Parameters

s	The string to trim.

Returns

The trimmed string.

8.5.2.19 std::string & pgutils::trim_front (std::string & *s*)

Trims leading whitespace from a string.

Parameters

s	The string to trim.

Returns

The trimmed string.

8.6 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 * progname = "gl_db"
  Static variable for program name.

8.6.1 Detailed Description

Administrative database management program.

8.6.2 Function Documentation

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

Checks if database, hostname and username were provided.

Parameters

config	Reference to a Config object.

Returns

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

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

Prints help or version messages if requested.

Parameters

config	Reference to a Config object.

Returns

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

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

Gets a password from the terminal.

Returns

The password.

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

Main function.

Parameters

argc	Number of command line arguments.
argv	Command line arguments.

Returns

Exit status code.

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

Sets program configuration options.

Parameters

config	Reference to a Config object.
argc	`argc` passed to `main()`.
argv	`argv` passed to `main()`.

8.7 Reporting program.

Functions

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

Variables

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

8.7.1 Detailed Description

Administrative reporting program.

8.7.2 Function Documentation

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

Checks if database, hostname and username were provided.

Parameters

config	Reference to a Config object.

Returns

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

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

Prints help or version messages if requested.

Parameters

config	Reference to a Config object.

Returns

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

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

Gets a password from the terminal.

Returns

The password.

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

Main function.

Parameters

argc	Number of command line arguments.
argv	Command line arguments.

Returns

Exit status code.

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

Sets program configuration options.

Parameters

config	Reference to a Config object.
argc	`argc` passed to `main()`.
argv	`argv` passed to `main()`.

8.8 User administration program.

Functions

- static void set_configuration (Config &config, int argc, char ∗argv[])

  Sets program configuration options.
- static bool check_help_and_version (const Config &config)

  Prints help or version messages if requested.
- static bool check_db_parameters (const Config &config)

  Checks if database, hostname and username were provided.
- GLUser get_user (Config &config, GLDatabase &gdb)

  Returns a user from either an ID or a name.
- static void show_user_details (const GLUser &user)

  Outputs details for a user.
- static void enable_user (GLUser &user, Config &config, GLDatabase &gdb)

  Enables or disables a user.
- static void set_user_password (GLUser &user, Config &config, GLDatabase &gdb)

  Sets a user's password.
- static void check_user_password (GLUser &user, Config &config)

  Checks a user's password.
- static void print_usage_message ()

  Prints a program usage message.
- static void print_version_message ()

  Prints a program version message.
- static void print_help_message ()

  Prints a program help message.
- static std::string login (void)

  Gets a password from the terminal.
- int main (int argc, char ∗argv[])

  Main function.

Variables

- static const char ∗ progname = "gl_user"

  Static variable for program name.

8.8.1 Detailed Description

User administration program.

8.8.2 Function Documentation

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

Checks if database, hostname and username were provided.

Parameters

config	Reference to a Config object.

Returns

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

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

Prints help or version messages if requested.

Parameters

config	Reference to a Config object.

Returns

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

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

Checks a user's password.

Parameters

user	Reference to user.
config	Reference to program configuration options.

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

Enables or disables a user.

Parameters

user	Reference to user.
config	Reference to program configuration.
gdb	Reference to database object.

8.8.2.5 GLUser get_user (Config & *config,* GLDatabase & *gdb*)

Returns a user from either an ID or a name.

Parameters

config	Program configurations object.
gdb	Database object.

Returns

The user.

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

Gets a password from the terminal.

Returns

The password.

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

Main function.

Parameters

argc	Number of command line arguments.
argv	Command line arguments.

Returns

Exit status code.

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

Sets program configuration options.

Parameters

config	Reference to a Config object.
argc	`argc` passed to `main()`.
argv	`argv` passed to `main()`.

8.8.2.9 static void set_user_password (GLUser & *user*, Config & *config*, GLDatabase & *gdb*) [static]

Sets a user's password.

Parameters

user	Reference to user.
config	Reference to program configuration.
gdb	Reference to database object.

8.8.2.10 static void show_user_details (const GLUser & *user*) [static]

Outputs details for a user.

Parameters

user	Reference to user.

Chapter 9

Class Documentation

9.1 genleg::Config Class Reference

Configuration options class.

`#include <config.h>`

Public Member Functions

- Config ()
- ~Config ()
- void add_cmdline_option (const std::string option, const enum Argument arg)

  Adds a supported command line option.
- void populate_from_cmdline (const int argc, char *const *argv)

  Populates options from the command line.
- void populate_from_file (const std::string filename)

  Populates options from a configuration file.
- bool is_set (const std::string option) const

  Checks is an option is set.
- const std::string & operator[] (const std::string &option) const

  operator[] overload.

Private Attributes

- std::map< std::string, std::string > m_opts_set
- std::list< std::pair < std::string, enum Argument > > m_opts_supp

9.1.1 Detailed Description

Configuration options class.

9.1.2 Constructor & Destructor Documentation

9.1.2.1 Config::Config ( )

Constructor

9.1.2.2 Config::~Config ( )

Destructor

9.1.3 Member Function Documentation

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

Adds a supported command line option.

Parameters

option	The name of the option.
arg	The argument specification for the option.

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

Checks is an option is set.

Parameters

option	The name of the option to check.

Returns

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

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

operator[] overload.

Retrieves the value of a set option.

Parameters

option	The name of the option.

Returns

The value of the option.

Exceptions

ConfigOptionNotSet	If the named option has not been set.

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

Populates options from the command line.

Parameters

argc	`argc` supplied to `main()`.
argv	`argv` supplied to `main()`.

Exceptions

ConfigBadOption	If an unsupported option is specified, or if a required argument is missing, or if an unexpected argument is found.

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

Populates options from a configuration file.

Parameters

filename	The name of the configuration file.

Exceptions

ConfigCouldNotOpenFile	If the configuration file cannot be opened.
ConfigBadConfigFile	If the configuration file is badly formed.

9.1.4 Member Data Documentation

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

Map of options which have been set

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

List of options which are supported

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

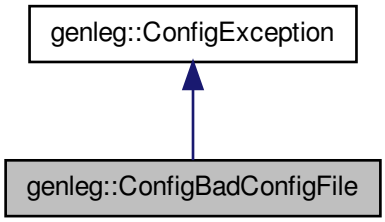
- lib/config/config.h
- lib/config/config.cpp
- lib/config/config_getopt.cpp

9.2 genleg::ConfigBadConfigFile Class Reference

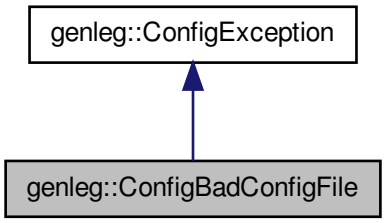
Exception class for badly formed configuration file.

`#include <config.h>`

Inheritance diagram for genleg::ConfigBadConfigFile:



Collaboration diagram for genleg::ConfigBadConfigFile:



Public Member Functions

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

9.2.1 Detailed Description

Exception class for badly formed configuration file.

9.2.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

msg	Database error message

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

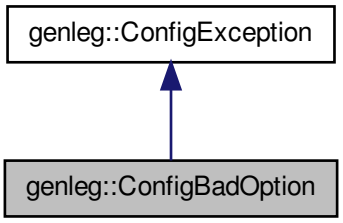
- lib/config/config.h

9.3 genleg::ConfigBadOption Class Reference

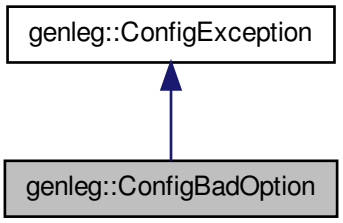
Exception class for bad provided option.

`#include <config.h>`

Inheritance diagram for genleg::ConfigBadOption:



Collaboration diagram for genleg::ConfigBadOption:



Public Member Functions

- ConfigBadOption (const std::string &msg)

  Constructor.

9.3.1 Detailed Description

Exception class for bad provided option.

9.3.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

msg	Database error message

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

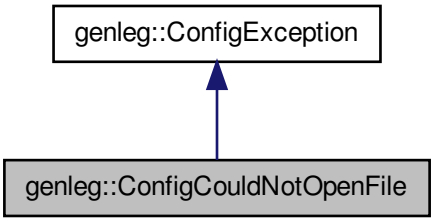
- lib/config/config.h

9.4 genleg::ConfigCouldNotOpenFile Class Reference

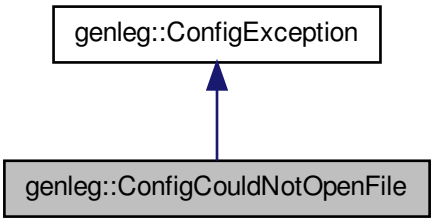
Exception class for when conf file cannot be opened.

`#include <config.h>`

Inheritance diagram for genleg::ConfigCouldNotOpenFile:



Collaboration diagram for genleg::ConfigCouldNotOpenFile:



Public Member Functions

- ConfigCouldNotOpenFile (const std::string &msg)

  Constructor.

9.4.1 Detailed Description

Exception class for when conf file cannot be opened.

9.4.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

msg	Database error message

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

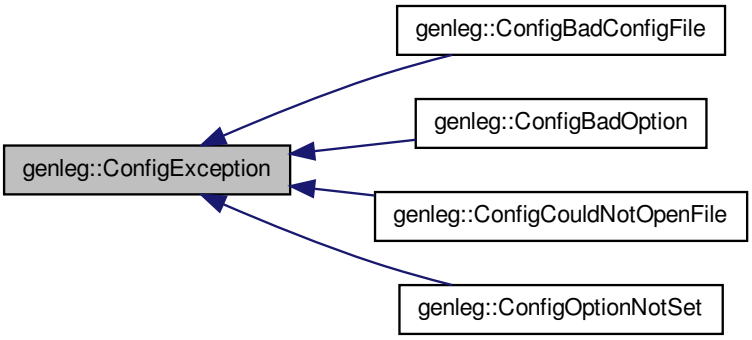
- lib/config/config.h

9.5 genleg::ConfigException Class Reference

Configuration module exception base class.

`#include <config.h>`

Inheritance diagram for genleg::ConfigException:



Public Member Functions

- ConfigException (const std::string &msg)

  Constructor.

9.5.1 Detailed Description

Configuration module exception base class.

9.5.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

msg	Database error message

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

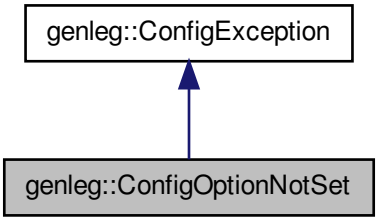
- lib/config/config.h

9.6 genleg::ConfigOptionNotSet Class Reference

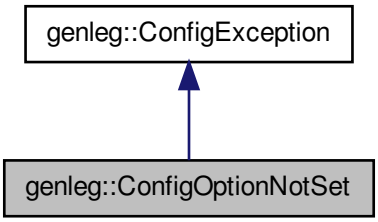
Exception class for option not set.

`#include <config.h>`

Inheritance diagram for genleg::ConfigOptionNotSet:



Collaboration diagram for genleg::ConfigOptionNotSet:



Public Member Functions

- ConfigOptionNotSet (const std::string &msg)

Constructor.

9.6.1 Detailed Description

Exception class for option not set.

9.6.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

msg	Database error message

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

- lib/config/config.h

9.7 pgutils::Currency Class Reference

Currency amount class.

`#include <currency.h>`

Public Member Functions

- Currency (const int64_t i=0, const uint8_t f=0)

  Constructor.
- Currency operator- () const

  Unary negation opertor.
- Currency & operator+= (const Currency &rhs)

  Addition assignment operator.
- Currency & operator-= (const Currency &rhs)

  Subtraction assignment operator.

Private Member Functions

- int64_t expand () const

  Returns a Currency amount as a whole integer.

Private Attributes

- int64_t m_int
- int m_frac

Friends

- bool operator== (const Currency &lhs, const Currency &rhs)

  Currency equality comparison operator.
- bool operator< (const Currency &lhs, const Currency &rhs)

  Currency less than comparison operator.
- Currency operator+ (Currency lhs, const Currency &rhs)

  Currency addition operator.

9.7.1 Detailed Description

Currency amount class.

9.7.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

i	The integer part.
f	The fractional part.

9.7.3 Member Function Documentation

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

Returns a Currency amount as a whole integer.

Returns

The Currency amount expanded to a whole integer, equivalent to multiplying it by 100.

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

Addition assignment operator.

Parameters

rhs	Right hand side currency amount.

Returns

A reference to the original currency amount.

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

Unary negation opertor.

Returns

The negated currency amount.

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

Subtraction assignment operator.

Parameters

rhs	Right hand side currency amount.

Returns

A reference to the original currency amount.

9.7.4 Friends And Related Function Documentation

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

Currency addition operator.

Friend addition operator function

Parameters

lhs	Left hand side.
rhs	Right hand side.

Returns

The sum of the two sides.

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

Currency less than comparison operator.

Friend less than comparison operator function

Parameters

lhs	Left hand side.
rhs	Right hand side.

Return values

true	If the lhs is less than the rhs.
false	If the lhs is not less than the rhs.

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

Currency equality comparison operator.

Friend equality operator function

Parameters

lhs	Left hand side.
rhs	Right hand side.

Return values

true	If the two sides are equal.
false	If the two sides are not equal.

9.7.5 Member Data Documentation

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

Fractional part

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

Integer part

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

- lib/pgutils/currency.h
- lib/pgutils/currency.cpp

9.8 pgutils::CurrencyException Class Reference

Base Currency exception class.

`#include <currency.h>`

Public Member Functions

- CurrencyException (const std::string &msg)

  Constructor.

9.8.1 Detailed Description

Base Currency exception class.

9.8.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

msg	Database error message

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

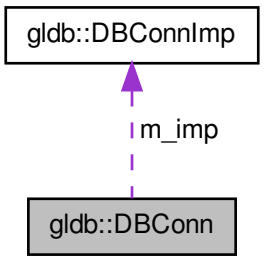
- lib/pgutils/currency.h

9.9 gldb::DBConn Class Reference

Database connection class.

`#include <dbconn.h>`

Collaboration diagram for gldb::DBConn:



Public Member Functions

- DBConn (DBConnImp *imp)

  Constructor.
- ~DBConn ()

  Destructor..
- void query (const std::string &sql_query)

  Runs an SQL query.
- Table select (const std::string &query)

  Runs an SQL SELECT query.
- void begin_transaction ()

  Begins a transaction.
- void rollback_transaction ()

  Rolls back a transaction.
- void commit_transaction ()

  Commits a transaction.
- unsigned long long last_auto_increment ()

  Returns the last auto incremented value.
- DBConn (const DBConn &)
- DBConn (const DBConn &&)
- DBConn & operator= (const DBConn &)
- DBConn & operator= (const DBConn &&)

Private Attributes

- DBConnImp * m_imp

9.9.1 Detailed Description

Database connection class.

9.9.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

imp	Pointer to database implementation object.

9.9.2.2 gldb::DBConn::DBConn (const DBConn &)

Deleted copy constructor

9.9.2.3 gldb::DBConn::DBConn (const DBConn &&)

Deleted move constructor

9.9.3 Member Function Documentation

9.9.3.1 unsigned long long DBConn::last_auto_increment ( )

Returns the last auto incremented value.

Returns

The last auto incremented value.

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

Deleted copy assignment operator

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

Deleted move assignment operator

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

Runs an SQL query.

Parameters

sql_query	The query.

Returns

A Table object containing the results.

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

Runs an SQL SELECT query.

Parameters

query	The query.

Returns

A Table object containing the results.

9.9.4 Member Data Documentation

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

Pointer to database implementation object.

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

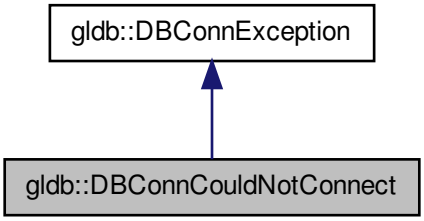
- lib/database/dbconn.h
- lib/database/dbconn.cpp

9.10 gldb::DBConnCouldNotConnect Class Reference

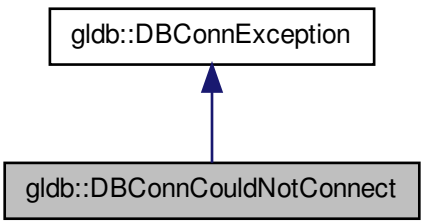
Could not connect to database exception class.

`#include <dbconn.h>`

Inheritance diagram for gldb::DBConnCouldNotConnect:



Collaboration diagram for gldb::DBConnCouldNotConnect:



Public Member Functions

- DBConnCouldNotConnect (const std::string &msg)

  Constructor.

9.10.1 Detailed Description

Could not connect to database exception class.

9.10.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

msg	Database error message

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

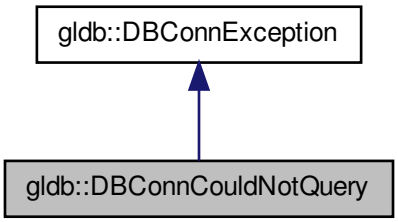
- lib/database/dbconn.h

9.11 gldb::DBConnCouldNotQuery Class Reference

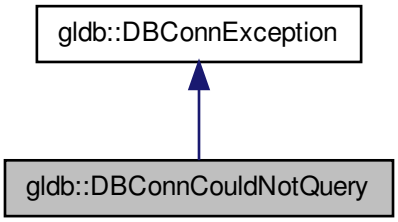
Could not execute database query exception class.

`#include <dbconn.h>`

Inheritance diagram for gldb::DBConnCouldNotQuery:



Collaboration diagram for gldb::DBConnCouldNotQuery:



Public Member Functions

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

9.11.1 Detailed Description

Could not execute database query exception class.

9.11.2 Constructor & Destructor Documentation

9.11.2.1 gldb::DBConnCouldNotQuery::DBConnCouldNotQuery (const std::string & *msg*) [inline], [explicit]

Constructor.

Parameters

msg	Database error message

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

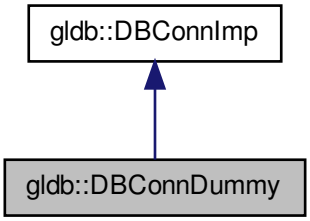
- lib/database/dbconn.h

9.12 gldb::DBConnDummy Class Reference

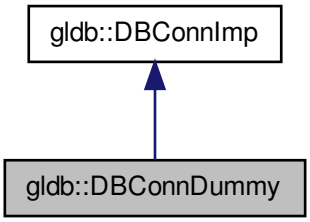
Dummy database implementation class.

`#include <dbconn_dummy_imp.h>`

Inheritance diagram for gldb::DBConnDummy:



Collaboration diagram for gldb::DBConnDummy:



Public Member Functions

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

  Constructor.
- DBConnDummy (const DBConnDummy &)
- virtual ~DBConnDummy ()
- DBConnDummy & operator= (const DBConnDummy &)
- virtual void query (const std::string &sql_query)

  Runs an SQL query.
- Table select (const std::string &query)

  Fakes running of an SQL SELECT query.

9.12.1 Detailed Description

Dummy database implementation class.

9.12.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

database	The name of the Dummy database.
hostname	The hostname of the server.
username	The username to log into the database.
password	The password to log into the database.

9.12.2.2 gldb::DBConnDummy::DBConnDummy (const **DBConnDummy** &)

Deleted copy constructor

9.12.2.3 DBConnDummy::~DBConnDummy ( ) [virtual]

Destructor

9.12.3 Member Function Documentation

9.12.3.1 DBConnDummy& gldb::DBConnDummy::operator= (const **DBConnDummy** &)

Deleted assignment operator

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

Runs an SQL query.

Parameters

sql_query	The query.

Exceptions

DBConnCouldNotQuery	If could not successfully execute query.

Implements gldb::DBConnImp.

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

Fakes running of an SQL SELECT query.

Parameters

query	Any query.

Returns

A Table object containing dummy results.

Implements gldb::DBConnImp.

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

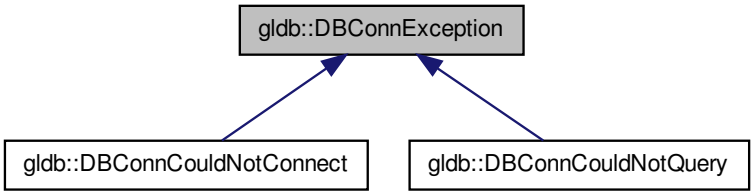
- lib/database_imp/dummy/dbconn_dummy_imp.h
- lib/database_imp/dummy/dbconn_dummy_imp.cpp

9.13 gldb::DBConnException Class Reference

Base database connection exception class.

`#include <dbconn.h>`

Inheritance diagram for gldb::DBConnException:



Public Member Functions

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

9.13.1 Detailed Description

Base database connection exception class.

9.13.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

msg	Database error message

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

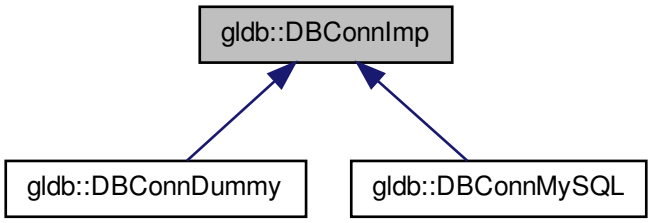
- lib/database/dbconn.h

9.14 gldb::DBConnImp Class Reference

Abstract database implementation base class.

`#include <dbconnimp.h>`

Inheritance diagram for gldb::DBConnImp:



Public Member Functions

- DBConnImp ()
- virtual ~DBConnImp ()
- virtual void query (const std::string &sql_query)=0

  Runs an SQL query.
- virtual Table select (const std::string &query)=0

  Runs an SQL SELECT query.
- virtual void begin_transaction ()=0

  Begins a transaction.
- virtual void rollback_transaction ()=0

  Rolls back a transaction.
- virtual void commit_transaction ()=0

  Commits a transaction.
- virtual unsigned long long last_auto_increment ()=0

  Returns the last auto incremented value.

9.14.1 Detailed Description

Abstract database implementation base class.

9.14.2 Constructor & Destructor Documentation

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

Constructor

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

Destructor

9.14.3 Member Function Documentation

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

Returns the last auto incremented value.

Returns

The last auto incremented value.

Implemented in gldb::DBConnMySQL.

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

Runs an SQL query.

Parameters

sql_query	The query.

Implemented in gldb::DBConnMySQL, and gldb::DBConnDummy.

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

Runs an SQL SELECT query.

Parameters

query	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:

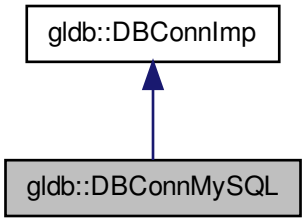
- lib/database/dbconnimp.h

9.15 gldb::DBConnMySQL Class Reference

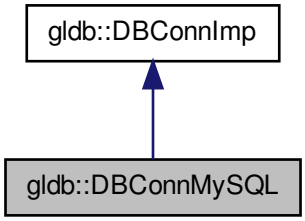
MySQL database implementation class.

`#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 &)
- DBConnMySQL (const DBConnMySQL &&)
- virtual ~DBConnMySQL ()
- DBConnMySQL & operator= (const DBConnMySQL &)
- DBConnMySQL & operator= (const DBConnMySQL &&)
- virtual void query (const std::string &sql_query)

  Runs an SQL query.
- virtual Table select (const std::string &sql_query)

  Runs an SQL SELECT query.
- virtual void begin_transaction ()

  Begins a transaction.
- virtual void rollback_transaction ()

  Rolls back a transaction.
- virtual void commit_transaction ()

  Commits a transaction.
- virtual unsigned long long last_auto_increment ()

  Returns the last auto incremented value.

Private Attributes

- MYSQL * m_conn

Static Private Attributes

- static std::mutex mtx

9.15.1 Detailed Description

MySQL database implementation class.

9.15.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

database	The name of the MySQL database.
hostname	The hostname of the server.
username	The username to log into the database.
password	The password to log into the database.

Exceptions

DBConnCouldNotConnect	If could not connect to database.

9.15.2.2 gldb::DBConnMySQL::DBConnMySQL (const DBConnMySQL &)

Deleted copy constructor

9.15.2.3 gldb::DBConnMySQL::DBConnMySQL (const DBConnMySQL &&)

Delete move constructor

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

Destructor

9.15.3 Member Function Documentation

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

Returns the last auto incremented value.

Returns

The last auto incremented value.

Implements gldb::DBConnImp.

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

Deleted assignment operator

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

Deleted move assignment operator

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

Runs an SQL query.

Parameters

sql_query	The SQL query.

Exceptions

DBConnCouldNotQuery	If could not successfully execute query.

Implements gldb::DBConnImp.

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

Runs an SQL SELECT query.

Parameters

sql_query	The SQL query.

Returns

A Table object containing the results.

Exceptions

DBConnCouldNotQuery	If could not successfully execute query.

Implements gldb::DBConnImp.

9.15.4 Member Data Documentation

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

The initialized MySQL handle.

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

Database connection mutex

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

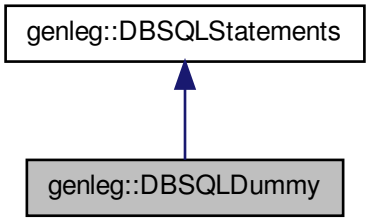
- lib/database_imp/mysql/dbconn_mysql_imp.h
- lib/database_imp/mysql/dbconn_mysql_imp.cpp

9.16 genleg::DBSQLDummy Class Reference

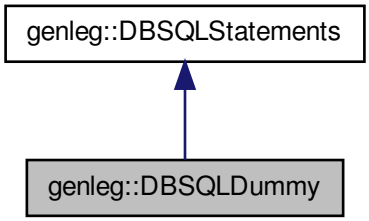
Dummy SQL statements class.

`#include <dbsql_dummy.h>`

Inheritance diagram for genleg::DBSQLDummy:



Collaboration diagram for genleg::DBSQLDummy:



Additional Inherited Members

9.16.1 Detailed Description

Dummy SQL statements class.

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

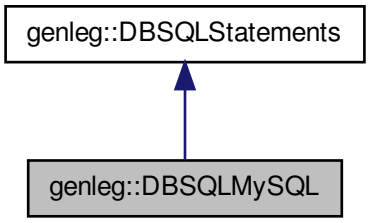
- lib/dbsql/dbsql_dummy.h

9.17 genleg::DBSQLMySQL Class Reference

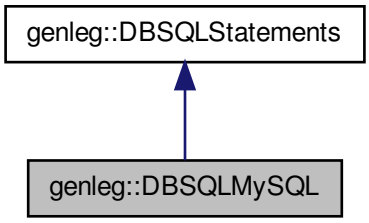
MySQL SQL statements class.

`#include <dbsql_mysql.h>`

Inheritance diagram for genleg::DBSQLMySQL:



Collaboration diagram for genleg::DBSQLMySQL:



Additional Inherited Members

9.17.1 Detailed Description

MySQL SQL statements class.

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

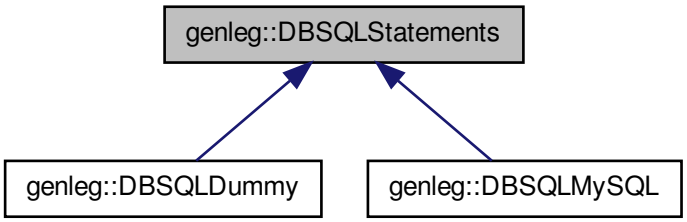
- lib/dbsql/dbsql_mysql.h

9.18 genleg::DBSQLStatements Class Reference

SQL statements class.

`#include <dbsqlstatements.h>`

Inheritance diagram for genleg::DBSQLStatements:



Public Member Functions

- DBSQLStatements ()
- virtual ~DBSQLStatements ()
- virtual std::string create_table (const std::string &table_name) const

  Returns a SQL statement for creating a table.
- virtual std::string drop_table (const std::string &table_name) const

  Returns a SQL statement for dropping a table.
- virtual std::string create_view (const std::string &view_name) const

  Returns a SQL statement for creating a view.
- virtual std::string drop_view (const std::string &view_name) const

  Returns a SQL statement for dropping a view.
- virtual std::string standing_data () const

  Returns a SQL statement to get the standing data.
- virtual std::string user_by_id (const std::string &user_id) const

  Returns a SQL statement to select a user by ID.
- virtual std::string user_by_username (const std::string &user_name) const

  Returns a SQL statement to select a user by username.
- virtual std::string update_user (const GLUser &user) const

  Returns a SQL UPDATE statement to update a user.
- virtual std::string entity_by_id (const std::string &entity_id) const

  Returns a SQL statement to select an entity by ID.
- virtual std::string entity_by_name (const std::string &entity_name) const

  Returns a SQL statement to select an entity by short name.
- virtual std::string account_by_name (const std::string &acc_name) const

  Returns a SQL statement to select a nominal account by name/number.
- virtual std::string je_by_id (const std::string &je_id) const

  Returns a SQL statement to select a journal entry by ID.
- virtual std::string jelines_by_id (const std::string &je_id) const

  Returns a SQL statement to select journal entry lines by ID.
- virtual std::string post_je (const unsigned int user, const unsigned int entity, const int period, const int year, const std::string &source, const std::string &memo) const

  Returns a SQL INSERT statement to post a journal entry.
- virtual std::string post_je_line (const unsigned long long je, const std::string account, const std::string amount) const

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

- virtual std::string grant (const std::string &user_id, const std::string &perm) const

  Returns a SQL statement to grant a user a permission.
- virtual std::string revoke (const std::string &user_id, const std::string &perm) const

  Returns a SQL UPDATE statement to revoke a permission from a user.
- virtual std::string get_perms (const std::string &user_id) const

  Returns a SQL UPDATE statement to list a user's permissions.
- virtual std::string currenttb () const

  Returns a SQL statement to run the current trial balance report.
- virtual std::string currenttb_by_entity (const std::string &entity) const

  Returns a SQL statement to run the current trial balance report by entity.
- std::string listusers () const

  Returns a SQL statement to run the list users report.

9.18.1 Detailed Description

SQL statements class.

9.18.2 Constructor & Destructor Documentation

9.18.2.1 DBSQLStatements::DBSQLStatements ( )

Constructor

9.18.2.2 DBSQLStatements::~DBSQLStatements ( ) [virtual]

Destructor

9.18.3 Member Function Documentation

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

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

Parameters

acc_name	The account name/number.

Returns

The SQL statement.

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

Returns a SQL statement for creating a table.

Parameters

table_name	The table to create.

Returns

The SQL statement to create the table.

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

Returns a SQL statement for creating a view.

Parameters

view_name	The view to create.

Returns

The SQL statement to create the view.

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

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

Returns

The SQL statement.

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

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

Parameters

entity	The entity number for which to run the report.

Returns

The SQL statement.

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

Returns a SQL statement for dropping a table.

Parameters

table_name	The table to drop.

Returns

The SQL statement to drop the table.

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

Returns a SQL statement for dropping a view.

Parameters

view_name	The view to drop.

Returns

The SQL statement to drop the view.

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

Returns a SQL statement to select an entity by ID.

Parameters

entity_id	The entity ID.

Returns

The SQL statement.

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

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

Parameters

entity_name	The entity short name.

Returns

The SQL statement.

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

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

Parameters

user_id	The user ID for which to list.

Returns

The SQL statement.

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

Returns a SQL statement to grant a user a permission.

Attention

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

Parameters

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

Returns

The SQL statement.

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

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

Parameters

je_id	The journal entry ID.

Returns

The SQL statement.

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

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

Parameters

je_id	The journal entry ID.

Returns

The SQL statement.

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

Returns a SQL statement to run the list users report.

Returns

The SQL statement.

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

Returns a SQL INSERT statement to post a journal entry.

Parameters

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

Returns

A string containing the query.

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

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

Parameters

je	The journal entry ID.
account	The account to which to post.
amount	The amount to post.

Returns

A string containing the SQL statement.

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

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

Parameters

user_id	The user ID from which to revoke.
perm	The permission to revoke.

Returns

The SQL statement.

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

Returns a SQL statement to get the standing data.

Returns

The SQL statement to get the data.

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

Returns a SQL UPDATE statement to update a user.

Parameters

user	A user object.

Returns

The SQL statement.

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

Returns a SQL statement to select a user by ID.

Parameters

user_id	The user_id

Returns

The SQL statement.

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

Returns a SQL statement to select a user by username.

Parameters

user_name	The username.

Returns

The SQL statement.

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

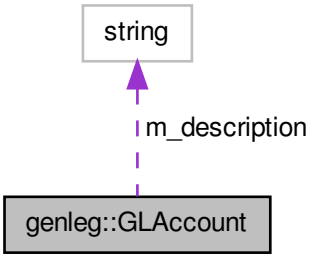
- lib/dbsql/dbsqlstatements.h
- lib/dbsql/dbsqlstatements.cpp

9.19 genleg::GLAccount Class Reference

Nominal account class.

`#include <glaccount.h>`

Collaboration diagram for genleg::GLAccount:



Public Member Functions

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

Constructor.

Public Attributes

- std::string m_description
- bool m_enabled

9.19.1 Detailed Description

Nominal account class.

9.19.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

number	The account number.
description	The account description.
enabled	The enabled flag.

9.19.3 Member Data Documentation

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

Account description

9.19.3.2 bool genleg::GLAccount::m_enabled

Enabled flag

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

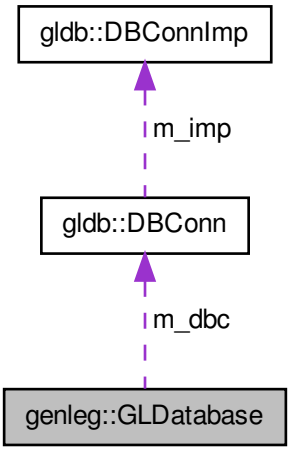
- lib/gldb/glaccount.h

9.20 genleg::GLDatabase Class Reference

General ledger database class.

`#include <gldatabase.h>`

Collaboration diagram for genleg::GLDatabase:



Public Member Functions

- GLDatabase (const std::string &database, const std::string &hostname, const std::string &username, const std::string &password)

  Constructor.
- ~GLDatabase ()
- void create_structure ()

  Creates the database structure.
- void destroy_structure ()

  Destroys the database structure.
- void load_sample_data (const std::string &dir)

  Loads sample data into the database.
- GLStandingData get_standing_data ()

  Gets the standing data.
- GLUser get_user_by_id (const std::string &user_id)

  Returns a user from an ID.
- GLUser get_user_by_username (const std::string &user_name)

  Returns a user from a user name.
- void update_user (const GLUser &user)

  Updates a user's details.
- void grant (const GLUser &user, const std::string &perm)

  Grants a user a permission.
- void revoke (const GLUser &user, const std::string &perm)

  Revokes a permission from a user.
- GLEntity get_entity_by_id (const std::string &entity_id)

  Returns an entity from an ID.
- GLEntity get_entity_by_name (const std::string &entity_name)

  Returns an entity from an entity short name.
- GLAccount get_account_by_name (const std::string &acc_name)

Returns a nominal account from an account number/name.

- GLJournal get_je_by_id (const std::string &je_id)

  Returns a journal entry from an ID.
- void post_journal (const GLJournal &journal)

  Posts a journal entry.
- GLReport report (const std::string &report_name, const std::string &arg="")

  Runs a report.

Static Public Member Functions

- static std::string backend ()

  Returns the backend database implementation.

Private Member Functions

- GLUser create_user (gldb::Table &table)

  Creates a user from a query table.
- GLEntity create_entity (gldb::Table &table)

  Creates an entity from a query table.
- GLReport standing_data_report ()

  Returns a standing data report.
- GLReport current_trial_balance_report (const std::string &entity)

  Returns a current trial balance report.
- GLReport list_users_report ()

  Returns a list users report.
- GLReport je_report (const std::string &je_id)

  Returns a single journal entry report.

Private Attributes

- gldb::DBConn m_dbc
- const std::shared_ptr< const DBSQLStatements > m_sql
- const std::vector< std::string > m_tables
- const std::vector< std::string > m_views

9.20.1 Detailed Description

General ledger database class.

9.20.2 Constructor & Destructor Documentation

9.20.2.1 GLDatabase::GLDatabase (const std::string & *database*, const std::string & *hostname*, const std::string & *username*, const std::string & *password*)

Constructor.

Parameters

database	Database name.
hostname	Hostname of database machine.
username	Username to log into database.
password	Password to log into database.

Exceptions

GLDBException	on error.

9.20.2.2 GLDatabase::~GLDatabase ( )

Destructor

9.20.3 Member Function Documentation

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

Returns the backend database implementation.

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

Returns

A string containing the database platform name.

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

Creates an entity from a query table.

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

Parameters

table	A table from the appropriate query.

Returns

The new entity.

9.20.3.3 void GLDatabase::create_structure ( )

Creates the database structure.

Exceptions

GLDBException	on error.

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

Creates a user from a query table.

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

Parameters

table	A table from the appropriate query.

Returns

The new user.

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

Returns a current trial balance report.

Parameters

entity	The entity for which to run the report, or an empty string for all entities.

Returns

A GLReport object with the report.

9.20.3.6 void GLDatabase::destroy_structure ( )

Destroys the database structure.

Exceptions

GLDBException	on error.

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

Returns a nominal account from an account number/name.

Parameters

acc_name	The account number/name.

Returns

The account.

Exceptions

GLDBException	if the user cannot be found.

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

Returns an entity from an ID.

Parameters

entity_id	The entity ID.

Returns

The entity.

Exceptions

GLDBException	if the entity cannot be found.

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

Returns an entity from an entity short name.

Parameters

entity_name	The entity short name.

Returns

The entity.

Exceptions

GLDBException	if the user cannot be found.

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

Returns a journal entry from an ID.

Parameters

je_id	The journal entry ID.

Returns

The entity.

9.20.3.11 GLStandingData GLDatabase::get_standing_data ( )

Gets the standing data.

Returns

The standing data.

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

Returns a user from an ID.

Parameters

user_id	The user ID.

Returns

The user.

Exceptions

GLDBException	if the user cannot be found.

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

Returns a user from a user name.

Parameters

user_name	The user name.

Returns

The user.

Exceptions

GLDBException	if the user cannot be found.

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

Grants a user a permission.

Parameters

user	The user for which to grant.
perm	A string containing the permission to grant.

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

Returns a single journal entry report.

Returns

A GLReport object with the report.

9.20.3.16 GLReport GLDatabase::list_users_report ( ) [private]

Returns a list users report.

Returns

A GLReport object with the report.

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

Loads sample data into the database.

Parameters

dir	The directory containing the sample data. Individual files in that directory should be named after the table they are intended to poplate.

Exceptions

GLDBException	on error.

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

Posts a journal entry.

Parameters

journal	The journal entry to post.

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

Runs a report.

Parameters

report_name	The name of the report.
arg	An optional argument.

Returns

A report object.

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

Revokes a permission from a user.

Parameters

user	The user for which to revoke.
perm	A string containing the permission to revoke.

9.20.3.21 GLReport GLDatabase::standing_data_report ( ) [private]

Returns a standing data report.

Returns

A GLReport object with the report.

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

Updates a user's details.

Parameters

user	The user object.

9.20.4 Member Data Documentation

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

Database connection

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

SQL statements object

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

Vector containing database table names

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

Vector containing database view names

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

- lib/gldb/gldatabase.h
- lib/gldb/gldatabase.cpp

9.21 genleg::GLDBException Class Reference

Base general ledger database exceptionc class.

`#include <glexception.h>`

Public Member Functions

- GLDBException (const std::string &msg)

  Constructor.

9.21.1 Detailed Description

Base general ledger database exceptionc class.

9.21.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

msg	Database error message

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

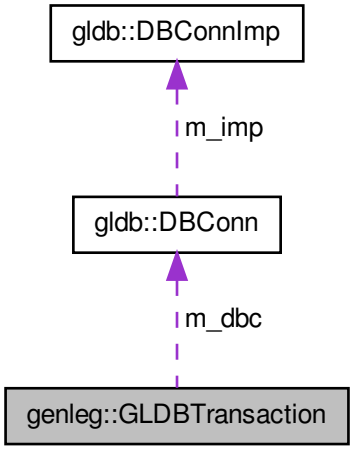
- lib/gldb/glexception.h

9.22 genleg::GLDBTransaction Class Reference

Database transaction RAII class.

`#include <gldatabase.h>`

Collaboration diagram for genleg::GLDBTransaction:



Public Member Functions

- GLDBTransaction (gldb::DBConn &dbc)

  Constructor.
- ~GLDBTransaction ()
- void commit ()

  Set commit flag.

Private Attributes

- gldb::DBConn & m_dbc
- bool m_commit

9.22.1 Detailed Description

Database transaction RAII class.

9.22.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

dbc	Database connection.

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

Destructor

9.22.3 Member Data Documentation

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

Commit flag

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

Database connection

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

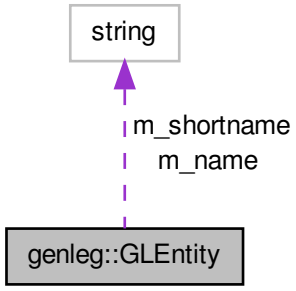
- lib/gldb/gldatabase.h

9.23 genleg::GLEntity Class Reference

General ledger entity class.

`#include <glentity.h>`

Collaboration diagram for genleg::GLEntity:



Public Member Functions

- GLEntity (const size_t id, const std::string &name, const std::string &shortname, const size_t parent, const bool aggregate, const bool enabled)

  Constructor.

Public Attributes

- std::string m_name
- std::string m_shortname
- const size_t m_parent
- const bool m_aggregate
- bool m_enabled

9.23.1 Detailed Description

General ledger entity class.

9.23.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

id	Entity ID.
name	Entity name.
shortname	Entity short name.
parent	Parent entity ID.
aggregate	Aggregate entity flag.
enabled	Enabled flag.

9.23.3 Member Data Documentation

9.23.3.1 const bool genleg::GLEntity::m_aggregate

Aggregate entity flag

9.23.3.2 bool genleg::GLEntity::m_enabled

Enabled flag

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

Entity name

9.23.3.4 const size_t genleg::GLEntity::m_parent

Parent entity ID

9.23.3.5 std::string genleg::GLEntity::m_shortname

Entity short name

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

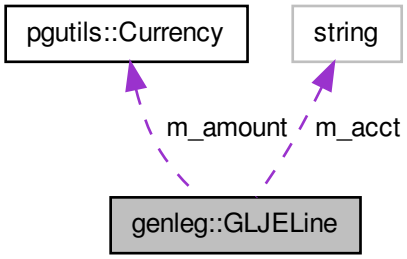
- lib/gldb/glentity.h

9.24 genleg::GLJELine Class Reference

Journal entry line class.

`#include <gljournal.h>`

Collaboration diagram for genleg::GLJELine:



Public Member Functions

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

Private Attributes

- std::string m_acct
- pgutils::Currency m_amount

9.24.1 Detailed Description

Journal entry line class.

9.24.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

account	The account name/ID
amount	The currency amount

9.24.3 Member Function Documentation

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

Returns the account name/number.

Returns

The account name/number.

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

Returns the currency amount.

Returns

The currency amount.

9.24.4 Member Data Documentation

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

Account number/name

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

Amount

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

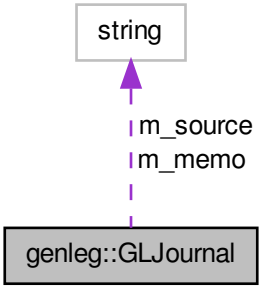
- lib/gldb/gljournal.h

9.25 genleg::GLJournal Class Reference

Journal entry class.

`#include <gljournal.h>`

Collaboration diagram for genleg::GLJournal:



Public Member Functions

- GLJournal (const unsigned long entity, const int period, const int year, const std::string &source, const std::string &memo, const size_t id=0, const size_t user=0)

  Constructor.
- iterator begin ()

  Returns an iterator to the first line.
- iterator end ()

  Returns an iterator to one past the last line.
- const_iterator begin () const

  Returns a const iterator to the first line.
- const_iterator end () const

  Returns a const iterator to one past the last line.

Private Attributes

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

9.25.1 Detailed Description

Journal entry class.

9.25.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

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

9.25.3 Member Function Documentation

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

Returns an iterator to the first line.

Returns

An iterator to the first line.

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

Returns a const iterator to the first line.

Returns

A const iterator to the first line.

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

Returns an iterator to one past the last line.

Returns

An iterator to one past the last line.

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

Returns a const iterator to one past the last line.

Returns

A const iterator to one past the last line.

9.25.4 Member Data Documentation

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

The entity number for the journal entry.

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

The journal entry ID

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

A vector of journal entry lines.

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

The memo for the journal entry.

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

The accounting period.

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

The journal entry source.

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

The journal entry posting user ID

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

The accounting year.

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

- lib/gldb/gljournal.h

9.26 genleg::GLReport Class Reference

General ledger report class.

`#include <glreport.h>`

Public Member Functions

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

Private Attributes

- const std::string m_title
- std::vector< std::pair < std::string, std::string > > m_headers
- const std::string m_report_text

9.26.1 Detailed Description

General ledger report class.

9.26.2 Constructor & Destructor Documentation

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

Constructor

9.26.3 Member Data Documentation

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

Report headers

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

The main report text

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

The report title

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

- lib/gldb/glreport.h

9.27 genleg::GLStandingData Class Reference

General ledger standing data class.

`#include <glstanding.h>`

Public Member Functions

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

Public Attributes

- int m_period
- int m_year
- int m_num_periods

9.27.1 Detailed Description

General ledger standing data class.

9.27.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

organization	The overall organization.
period	The current accounting period.
year	The current accounting year.
num_periods	The number of accounting periods in a year.

9.27.3 Member Data Documentation

9.27.3.1 int genleg::GLStandingData::m_num_periods

Number of periods per year

9.27.3.2 int genleg::GLStandingData::m_period

Current period

9.27.3.3 int genleg::GLStandingData::m_year

Current year

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

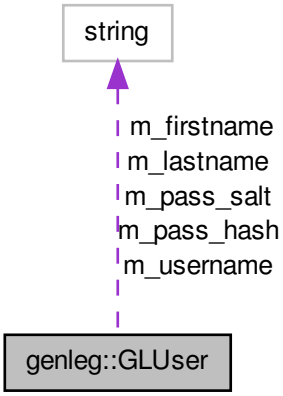
- lib/gldb/glstanding.h

9.28 genleg::GLUser Class Reference

General ledger user class.

`#include <gluser.h>`

Collaboration diagram for genleg::GLUser:



Public Member Functions

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

  Constructor.
- ~GLUser ()
- const std::string & id () const

  Returns the user ID.
- const std::string & username () const

  Returns the username.
- const std::string & firstname () const

  Returns the user's first name.
- const std::string & lastname () const

Returns the user's last name.

- const std::string & pass_hash () const

  Returns the user's hashed password.

- const std::string & pass_salt () const

  Returns the user's password salt.

- const std::vector< std::string > & permissions () const

  Returns the permissions for a user.

- bool enabled () const

  Returns the user's enabled status.

- void set_username (const std::string &new_username)

  Sets a user's username.

- void set_firstname (const std::string &new_firstname)

  Sets a user's first name.

- void set_lastname (const std::string &new_lastname)

  Sets a user's last name.

- void set_enabled (const bool new_enabled)

  Sets a user's enabled status.

- void set_password (const std::string &new_pass)

  Sets a user's password hash and salt.

- bool check_password (const std::string &check_pass)

  Checks a password against the user's hash.

Private Attributes

- const std::string m_id
- std::string m_username
- std::string m_firstname
- std::string m_lastname
- std::string m_pass_hash
- std::string m_pass_salt
- const std::vector< std::string > m_perms
- bool m_enabled

9.28.1 Detailed Description

General ledger user class.

9.28.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

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

9.28.2.2 GLUser::~GLUser ( )

Destructor

9.28.3 Member Function Documentation

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

Checks a password against the user's hash.

Parameters

check_pass	The password to check, must be $>$ 8 characters.

Returns

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

9.28.3.2 bool GLUser::enabled ( ) const

Returns the user's enabled status.

Returns

The user's enabled status.

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

Returns the user's first name.

Returns

The user's first name.

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

Returns the user ID.

Returns

The user ID.

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

Returns the user's last name.

Returns

The user's last name.

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

Returns the user's hashed password.

Returns

The user's hashed password.

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

Returns the user's password salt.

Returns

The user's password salt.

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

Returns the permissions for a user.

Returns

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

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

Sets a user's enabled status.

Parameters

new_enabled	The user's new enabled status.

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

Sets a user's first name.

Parameters

new_firstname	The user's new first name.

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

Sets a user's last name.

Parameters

new_lastname	The user's new last name.

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

Sets a user's password hash and salt.

Parameters

new_pass	The new password, must be $>$ 8 characters.

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

Sets a user's username.

Parameters

new_username	The user's new username.

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

Returns the username.

Returns

The username.

9.28.4 Member Data Documentation

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

User's enabled status

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

User's first name

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

User ID

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

User's last name

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

User's hashed password

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

User's password salt

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

List of permissions

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

Username

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

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

9.29 gldb::MySQLResult Class Reference

MySQL result structure class.

`#include <dbconn_mysql_result.h>`

Public Member Functions

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

Private Attributes

- MYSQL_RES * m_result
- unsigned int m_num_fields

9.29.1 Detailed Description

MySQL result structure class.

9.29.2 Constructor & Destructor Documentation

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

Constructor.

Parameters

conn	MySQL connection

Exceptions

DBConnCouldNotQuery	on failure

9.29.2.2 gldb::MySQLResult::~MySQLResult ( )

Destructor

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

Deleted copy constructor

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

Deleted move constructor

9.29.3 Member Function Documentation

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

Returns the number of fields in the result set.

Returns

The number of fields in the result set.

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

Deleted copy assignment operator

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

Deleted move assignment operator

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

Returns the MYSQL_RES pointer.

Returns

The MYSQL_RES pointer.

9.29.4 Member Data Documentation

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

The number of fields in the result set

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

The MYSQL_RES pointer

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

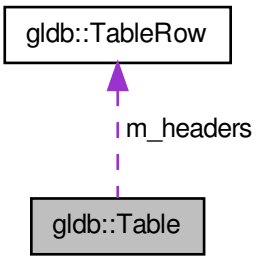
- lib/database_imp/mysql/dbconn_mysql_result.h
- lib/database_imp/mysql/dbconn_mysql_result.cpp

9.30 gldb::Table Class Reference

Database table class.

`#include <table.h>`

Collaboration diagram for gldb::Table:



Public Member Functions

- Table (const TableRow &headers)
  Constructor.
- Table (TableRow &&headers)
  Constructor with move semantics.
- Table (const Table &table)
  Copy constructor.
- Table (Table &&table)
  Move constructor.
- Table & operator= (const Table &table)
  Copy assignment operator.
- Table & operator= (Table &&table)
  Move assignment operator.
- ~Table ()
- size_t num_fields () const
  Returns the number of fields in each row.
- size_t num_records () const
  Returns the number of record in the table.
- iterator begin ()
  Returns iterator for beginning.
- iterator end ()
  Returns iterator for end plus one.
- const_iterator begin () const
  Returns const iterator for beginning.
- const_iterator end () const
  Returns const iterator for end plus one.
- void set_quoted (const std::vector< bool > &vec)
  Sets the quote flags for the records.
- void set_quoted (std::vector< bool > &&vec)

Sets the quote flags for the records with move semantics.

- const TableRow & get_headers () const

  Returns the field names.
- const TableRow & operator[] (const size_t idx) const

  Overloaded index operator.
- void append_record (const TableRow &new_record)

  Appends a record to the table.
- void append_record (TableRow &&new_record)

  Appends a record to the table with move semantics.
- std::string insert_query (const std::string &table_name, const size_t idx)

  Creates an SQL INSERT query from a table record.
- std::string get_field (const std::string &field_name, const size_t row_index)

  Gets a field from a record by field name.

Static Public Member Functions

- static Table create_from_file (const std::string &filename, const char delim)

  Creates a table from an input file.

Private Attributes

- TableRow m_headers
- std::vector< TableRow > m_records
- std::vector< bool > m_quoted

9.30.1 Detailed Description

Database table class.

9.30.2 Constructor & Destructor Documentation

9.30.2.1 Table::Table (const TableRow & *headers*) [explicit]

Constructor.

Parameters

headers	Table row containing field names.

9.30.2.2 Table::Table (TableRow && *headers*) [explicit]

Constructor with move semantics.

Parameters

headers	Table row containing field names.

9.30.2.3 Table::Table (const Table & *table*)

Copy constructor.

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

Parameters

table	Table to copy.

9.30.2.4 Table::Table (Table && *table*)

Move constructor.

Parameters

table	Table to move.

9.30.2.5 Table::~Table ( )

Destructor

9.30.3 Member Function Documentation

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

Appends a record to the table.

Parameters

new_record	The record to append.

9.30.3.2 void Table::append_record (TableRow && *new_record*)

Appends a record to the table with move semantics.

Parameters

new_record	The record to append.

9.30.3.3 iterator gldb::Table::begin ( ) [inline]

Returns iterator for beginning.

Returns

Iterator for beginning.

9.30.3.4 const_iterator gldb::Table::begin ( ) const [inline]

Returns const iterator for beginning.

Returns

Const iterator for beginning.

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

Creates a table from an input file.

Parameters

filename	The name of the input file.
delim	The delimiting character.

Returns

The table.

Exceptions

TableBadInputFile	on badly formed input file.
TableCouldNotOpenInput-File	on bad filename.

9.30.3.6 iterator gldb::Table::end ( ) [inline]

Returns iterator for end plus one.

Returns

Iterator for end plus one.

9.30.3.7 const_iterator gldb::Table::end ( ) const [inline]

Returns const iterator for end plus one.

Returns

Const iterator for end plus one.

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

Gets a field from a record by field name.

Parameters

field_name	The name of the field.
row_index	The index of the row.

Returns

The contents ofthe field.

Exceptions

TableNoSuchField	if `field_name` is not a valid field name.
TableNoSuchRecord	if there is no record at index `row_index`.

9.30.3.9 const TableRow& gldb::Table::get_headers ( ) const [inline]

Returns the field names.

Returns

The field names.

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

Creates an SQL INSERT query from a table record.

Parameters

table_name	The name of the table into which to INSERT.
idx	The index of the record.

Returns

A string containing the query.

9.30.3.11 size_t gldb::Table::num_fields ( ) const [inline]

Returns the number of fields in each row.

Returns

The number of fields in each row.

9.30.3.12 size_t gldb::Table::num_records ( ) const [inline]

Returns the number of record in the table.

Returns

The number of records in the table.

9.30.3.13 Table & Table::operator= (const Table & *table*)

Copy assignment operator.

Parameters

table	Table to copy.

Returns

Reference to the assigned-to table.

9.30.3.14 Table & Table::operator= (Table && *table*)

Move assignment operator.

Parameters

table	Table to move.

Returns

Reference to the assigned-to table.

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

Overloaded index operator.

Parameters

idx	The zero-based index of the record.

Returns

The selected record.

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

Sets the quote flags for the records.

Parameters

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

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

Sets the quote flags for the records with move semantics.

Parameters

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

9.30.4 Member Data Documentation

9.30.4.1 TableRow gldb::Table::m_headers [private]

The names of the fields

9.30.4.2 std::vector<bool> gldb::Table::m_quoted [private]

A vector to show if fields should be quoted for INSERT

9.30.4.3 std::vector<TableRow> gldb::Table::m_records [private]

A vector of the records

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

- lib/database/table.h

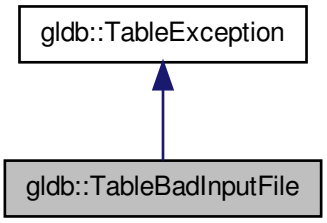
- lib/database/table.cpp

9.31 gldb::TableBadInputFile Class Reference

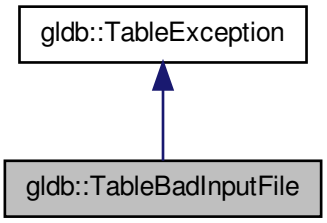
Could not connect to database exception class.

`#include <table.h>`

Inheritance diagram for gldb::TableBadInputFile:



Collaboration diagram for gldb::TableBadInputFile:



Public Member Functions

- TableBadInputFile (const std::string &msg)

  Constructor.

9.31.1 Detailed Description

Could not connect to database exception class.

9.31.2 Constructor & Destructor Documentation

9.31.2.1 gldb::TableBadInputFile::TableBadInputFile (const std::string & *msg*) [inline],[explicit]

Constructor.

Parameters

msg	Database error message

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

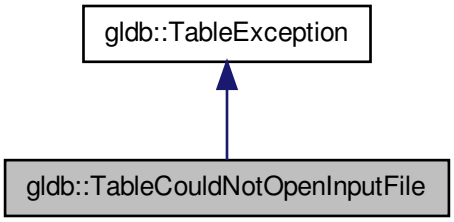
- lib/database/table.h

9.32 gldb::TableCouldNotOpenInputFile Class Reference

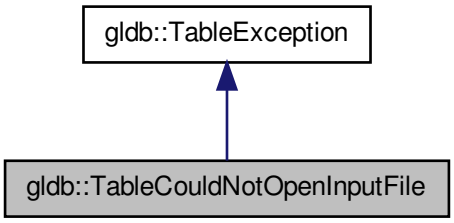
Could not connect to database exception class.

`#include <table.h>`

Inheritance diagram for gldb::TableCouldNotOpenInputFile:



Collaboration diagram for gldb::TableCouldNotOpenInputFile:



Public Member Functions

- TableCouldNotOpenInputFile (const std::string &msg)

  Constructor.

9.32.1 Detailed Description

Could not connect to database exception class.

9.32.2 Constructor & Destructor Documentation

9.32.2.1 gldb::TableCouldNotOpenInputFile::TableCouldNotOpenInputFile (const std::string & *msg*) [inline], [explicit]

Constructor.

Parameters

msg	Database error message

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

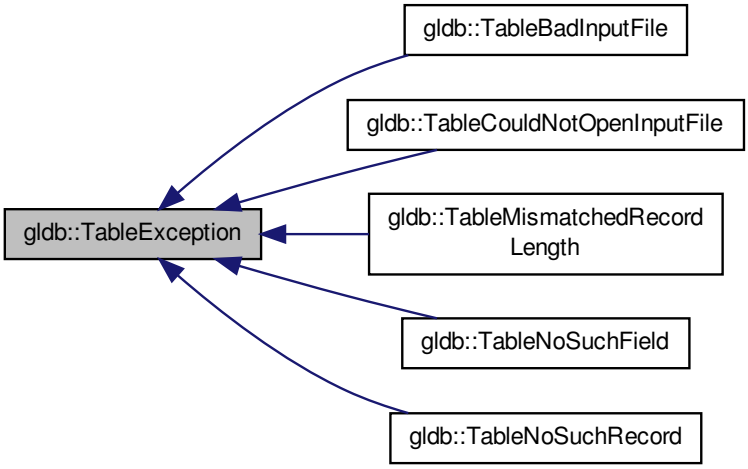
- lib/database/table.h

9.33 gldb::TableException Class Reference

Base database connection exception class.

`#include <table.h>`

Inheritance diagram for gldb::TableException:



Public Member Functions

- TableException (const std::string &msg)

  Constructor.

9.33.1 Detailed Description

Base database connection exception class.

9.33.2 Constructor & Destructor Documentation

9.33.2.1 gldb::TableException::TableException (const std::string & *msg*) [inline], [explicit]

Constructor.

Parameters

msg	Database error message

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

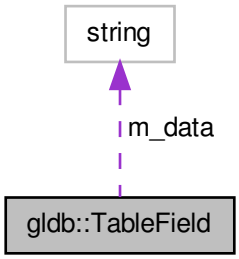
- lib/database/table.h

9.34 gldb::TableField Class Reference

Database table field class.

`#include <tablefield.h>`

Collaboration diagram for gldb::TableField:



Public Member Functions

- TableField (const char *data)

  Constructor accepting `const char *` data.
- TableField (const std::string &data)

  Constructor accepting `std:string` data.
- TableField (std::string &&data)

  Constructor accepting `std:string` data with move semantics.
- TableField (const TableField &field)

  Copy constructor.
- TableField (TableField &&field)

  Move constructor.

- ~TableField ()
- size_t length () const

  Returns the length of the field.
- operator std::string () const

  Overridden conversion operator.
- TableField & operator= (const char *data)

  Overridden assignment operator for `const char *`.
- TableField & operator= (const std::string &data)

  Overridden assignment operator for `std::string`.
- TableField & operator= (std::string &&data)

  Overridden assignment operator for `std::string` with move semantics.
- TableField & operator= (const TableField &field)

  Overridden copy assignment operator.
- TableField & operator= (TableField &&field)

  Overridden move assignment operator.
- char & operator[] (const size_t idx)

  Overridden index operator.
- const char & operator[] (const size_t idx) const

  Overridden index operator.
- TableField & operator+= (const char c)

  Overridden compound assignment operator.
- TableField & operator+= (const std::string &data)

  Overridden compound assignment operator.

Private Attributes

- std::string m_data

Friends

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

  Overridden << operator for printing a field.

9.34.1 Detailed Description

Database table field class.

9.34.2 Constructor & Destructor Documentation

9.34.2.1 TableField::TableField (const char * *data*) [explicit]

Constructor accepting `const char *` data.

Parameters

data	The initial contents of the field.

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

Constructor accepting `std:string` data.

Parameters

data	The initial contents of the field.

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

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

Parameters

data	The initial contents of the field.

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

Copy constructor.

Parameters

field	The field from which to copy.

9.34.2.5 TableField::TableField (TableField && *field*)

Move constructor.

Parameters

field	The field from which to move.

9.34.2.6 TableField::~TableField ( )

Destructor

9.34.3 Member Function Documentation

9.34.3.1 size_t gldb::TableField::length ( ) const [inline]

Returns the length of the field.

Returns

The length of the field.

9.34.3.2 gldb::TableField::operator std::string ( ) const [inline]

Overridden conversion operator.

Returns the field contents as a string.

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

Overridden compound assignment operator.

Parameters

c	The character to append to the field.

Returns

A reference to the same field.

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

Overridden compound assignment operator.

Parameters

data	The string to append to the field.

Returns

A reference to the same field.

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

Overridden assignment operator for `const char *`.

Parameters

data	The new contents of the field.

Returns

A reference to the same field.

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

Overridden assignment operator for `std::string`.

Parameters

data	The new contents of the field.

Returns

A reference to the same field.

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

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

Parameters

data	The new contents of the field.

Returns

A reference to the same field.

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

Overridden copy assignment operator.

Parameters

field	The field to copy.

Returns

A reference to the same field.

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

Overridden move assignment operator.

Parameters

field	The field to move.

Returns

A reference to the same field.

9.34.3.10 char& gldb::TableField::operator[] (const size_t *idx*) [inline]

Overridden index operator.

Parameters

idx	The desired index.

Returns

A reference to the character at the specified index.

9.34.3.11 const char& gldb::TableField::operator[] (const size_t *idx*) const [inline]

Overridden index operator.

Parameters

idx	The desired index.

Returns

A const reference to the character at the specified index.

9.34.4 Friends And Related Function Documentation

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

Overridden << operator for printing a field.

Parameters

out	The ostream to which to print.
field	A reference to the field.

Returns

A reference to `out`.

9.34.5 Member Data Documentation

9.34.5.1 std::string gldb::TableField::m_data [private]

The field contents

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

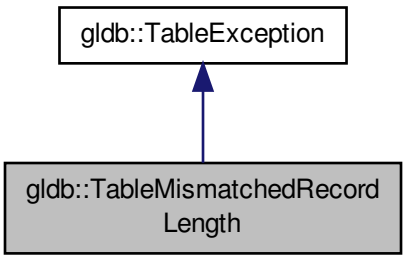
- lib/database/tablefield.h
- lib/database/tablefield.cpp

9.35 gldb::TableMismatchedRecordLength Class Reference

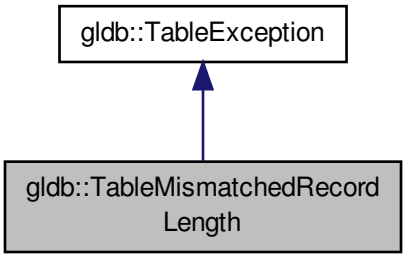
Mismatched record length exception class.

`#include <table.h>`

Inheritance diagram for gldb::TableMismatchedRecordLength:



Collaboration diagram for gldb::TableMismatchedRecordLength:



Public Member Functions

- TableMismatchedRecordLength (const std::string &msg)

  Constructor.

9.35.1 Detailed Description

Mismatched record length exception class.

9.35.2 Constructor & Destructor Documentation

9.35.2.1 gldb::TableMismatchedRecordLength::TableMismatchedRecordLength (const std::string & *msg*) [inline], [explicit]

Constructor.

Parameters

msg	Database error message

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

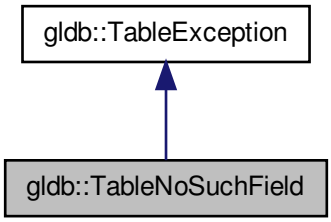
- lib/database/table.h

9.36 gldb::TableNoSuchField Class Reference

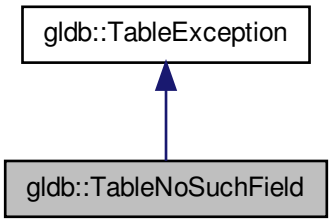
No such field exception class.

`#include <table.h>`

Inheritance diagram for gldb::TableNoSuchField:



Collaboration diagram for gldb::TableNoSuchField:



Public Member Functions

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

9.36.1 Detailed Description

No such field exception class.

9.36.2 Constructor & Destructor Documentation

9.36.2.1 gldb::TableNoSuchField::TableNoSuchField (const std::string & *msg*) [inline], [explicit]

Constructor.

Parameters

msg	Database error message

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

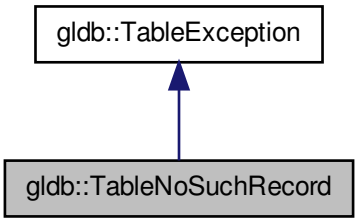
- lib/database/table.h

9.37 gldb::TableNoSuchRecord Class Reference

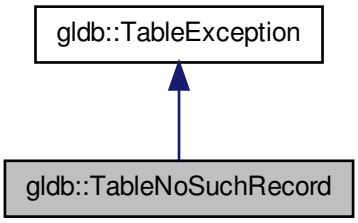
No such record exception class.

`#include <table.h>`

Inheritance diagram for gldb::TableNoSuchRecord:



Collaboration diagram for gldb::TableNoSuchRecord:



Public Member Functions

- TableNoSuchRecord (const std::string &msg)

  Constructor.

9.37.1 Detailed Description

No such record exception class.

9.37.2 Constructor & Destructor Documentation

9.37.2.1 gldb::TableNoSuchRecord::TableNoSuchRecord (const std::string & *msg*) [inline], [explicit]

Constructor.

Parameters

msg	Database error message

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

- lib/database/table.h

9.38 gldb::TableRow Class Reference

Database table row class.

`#include <tablerow.h>`

Public Member Functions

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

- void append_field (const char *new_field)

  Appends a field to the row.
- void append_field (const std::string &new_field)

  Appends a field to the row.
- void append_field (std::string &&new_field)

  Appends a field to the row with move semantics.
- void append_field (const TableField &new_field)

  Appends a field to the row.
- void append_field (TableField &&new_field)

  Appends a field to the row with move semantics.
- void print (std::ostream &stream) const

  Prints a row.
- std::string record_string (const std::vector< bool > &quoted) const

  Creates a comma separated string of fields.
- std::string record_string () const

  Creates an unquoted comma separated string of fields.

Private Attributes

- std::vector< TableField > m_fields

9.38.1 Detailed Description

Database table row class.

9.38.2 Constructor & Destructor Documentation

9.38.2.1 TableRow::TableRow ( )

Default constructor

9.38.2.2 TableRow::TableRow (const size_t *size*) [explicit]

Constructor with initial number of fields.

Parameters

size	The initial number of fields.

9.38.2.3 TableRow::TableRow (const std::vector< std::string > & *vec*) [explicit]

Constructor with string vector.

Parameters

vec	The vector.

9.38.2.4 TableRow::TableRow (std::vector< std::string > && *vec*) [explicit]

Constructor with string vector and move semantics.

Parameters

vec	The vector.

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

Constructor with std::string initializer list.

Parameters

i	The initializer list.

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

Copy constructor.

Parameters

row	The row to copy.

9.38.2.7 TableRow::TableRow (TableRow && *row*)

Move constructor.

Parameters

row	The row to move.

9.38.2.8 TableRow::~TableRow ( )

Destructor

9.38.3 Member Function Documentation

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

Appends a field to the row.

Parameters

new_field	The contents of the new field.

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

Appends a field to the row.

Parameters

new_field	The contents of the new field.

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

Appends a field to the row with move semantics.

Parameters

new_field	The contents of the new field.

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

Appends a field to the row.

Parameters

new_field	A field from which to copy.

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

Appends a field to the row with move semantics.

Parameters

new_field	A field from which to copy.

9.38.3.6 iterator gldb::TableRow::begin ( ) [inline]

Returns iterator for beginning.

Returns

Iterator for beginning.

9.38.3.7 const_iterator gldb::TableRow::begin ( ) const [inline]

Returns const iterator for beginning.

Returns

Const iterator for beginning.

9.38.3.8 iterator gldb::TableRow::end ( ) [inline]

Returns iterator for end plus one.

Returns

Iterator for end plus one.

9.38.3.9 const_iterator gldb::TableRow::end ( ) const [inline]

Returns const iterator for end plus one.

Returns

Const iterator for end plus one.

9.38.3.10 TableRow & TableRow::operator= (const TableRow & *row*)

Copy assignment operator.

Parameters

row	The row to copy.

Returns

A reference to the assigned-to row.

9.38.3.11 TableRow & TableRow::operator= (TableRow && *row*)

Move assignment operator.

Parameters

row	The row to move.

Returns

A reference to the assigned-to row.

9.38.3.12 TableField& gldb::TableRow::operator[] (const size_t *idx*) [inline]

Overridden index operator.

Parameters

idx	The zero-based index of the field.

Returns

A reference to the field at the specified index.

9.38.3.13 const TableField& gldb::TableRow::operator[] (const size_t *idx*) const [inline]

Overridden index operator.

Parameters

idx	The zero-based index of the field.

Returns

A const reference to the field at the specified index.

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

Prints a row.

Parameters

stream	The ostream to which to print.

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

Creates a comma separated string of fields.

Parameters

quoted	A vector of `bool`, for each field `true` means that field will be enclosed in single quotes in the comma separated string, `false` means it will not be.

Returns

The comma separated string.

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

Creates an unquoted comma separated string of fields.

Returns

The unquoted comma separated string.

9.38.3.17 size_t gldb::TableRow::size ( ) const [inline]

Returns the number of fields.

Returns

The number of fields.

9.38.4 Member Data Documentation

9.38.4.1 std::vector<TableField> gldb::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 10

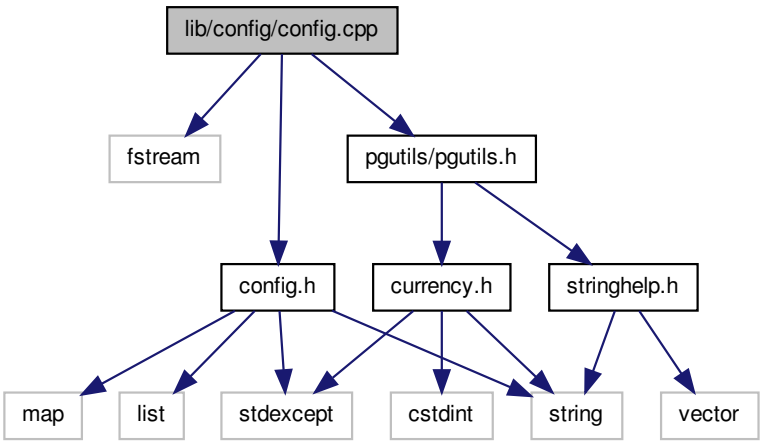
File Documentation

10.1 lib/config/config.cpp File Reference

Implementation of program configurations class.

```
#include <fstream>
#include "config.h"
#include "pgutils/pgutils.h"
```

Include dependency graph for config.cpp:



10.1.1 Detailed Description

Implementation of program configurations class.

Author

Paul Griffiths

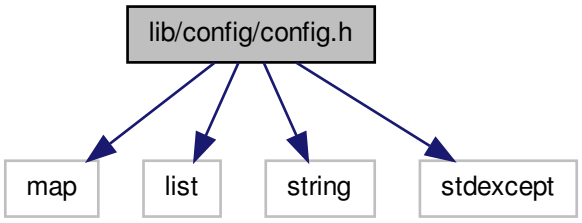
Copyright

10.2 lib/config/config.h File Reference

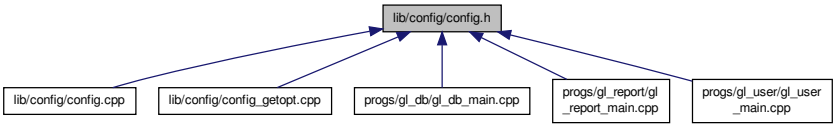
Interface to program configurations class.

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

Include dependency graph for config.h:



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



Classes

- class genleg::ConfigException

  Configuration module exception base class.
- class genleg::ConfigOptionNotSet

  Exception class for option not set.
- class genleg::ConfigBadOption

  Exception class for bad provided option.
- class genleg::ConfigCouldNotOpenFile

  Exception class for when conf file cannot be opened.
- class genleg::ConfigBadConfigFile

  Exception class for badly formed configuration file.
- class genleg::Config

  Configuration options class.

Enumerations

- enum genleg::Argument

  Enumeration class for option argument specifications.

10.2.1 Detailed Description

Interface to program configurations class.

Author

Paul Griffiths

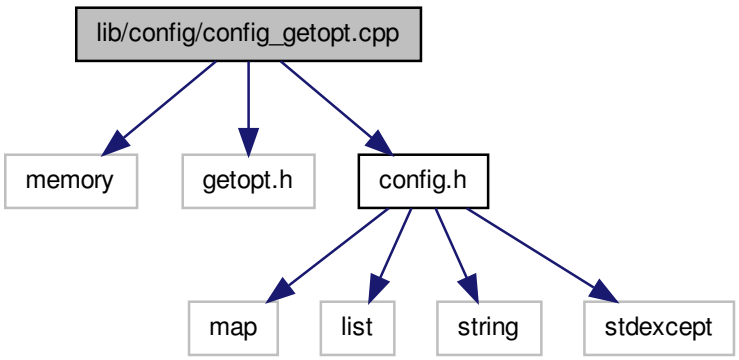
Copyright

10.3 lib/config/config_getopt.cpp File Reference

Implementation of command line functionality.

```
#include <memory>
#include <getopt.h>
#include "config.h"
```

Include dependency graph for config_getopt.cpp:



Macros

- #define _XOPEN_SOURCE 600

10.3.1 Detailed Description

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

Author

Paul Griffiths

Copyright

10.3.2 Macro Definition Documentation

10.3.2.1 #define _XOPEN_SOURCE 600

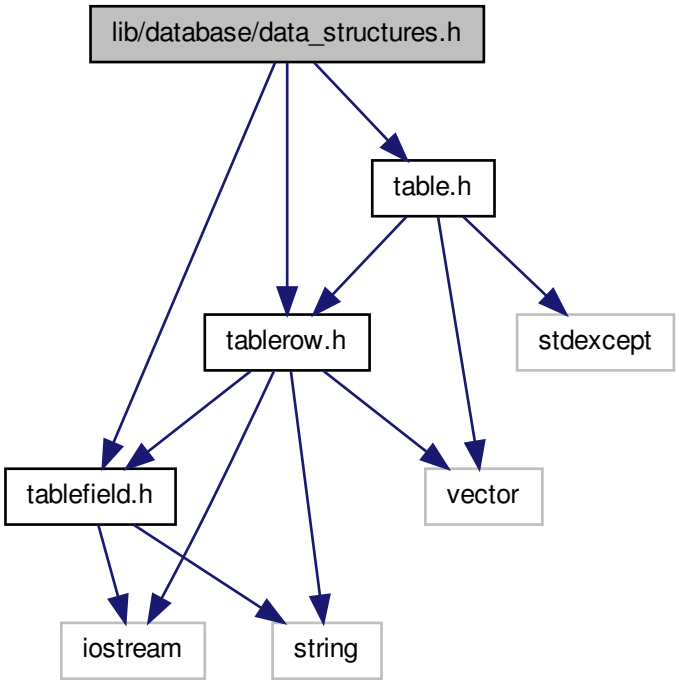
UNIX feature test macro for getopt library

10.4 lib/database/data_structures.h File Reference

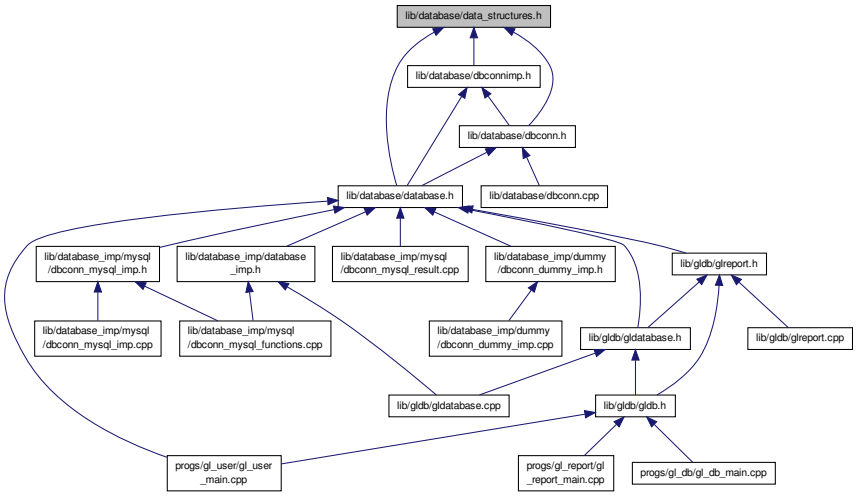
Main interface to database data structures.

```
#include "tablefield.h"
#include "tablerow.h"
#include "table.h"
```

Include dependency graph for data_structures.h:



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



10.4.1 Detailed Description

Main interface to database data structures.

Author

Paul Griffiths

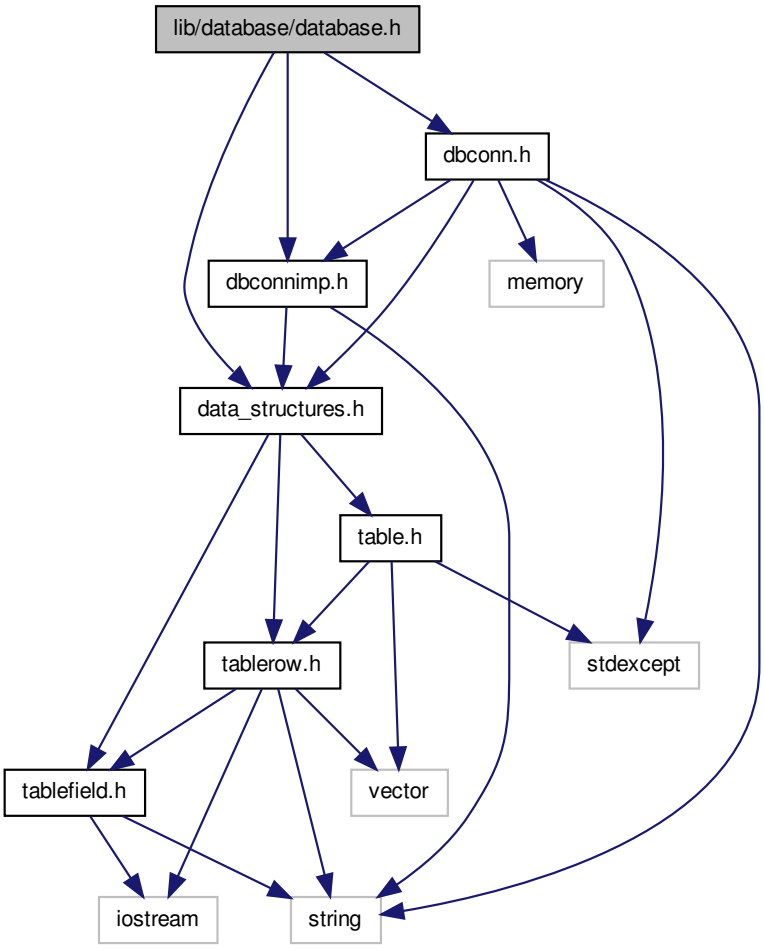
Copyright

10.5 lib/database/database.h File Reference

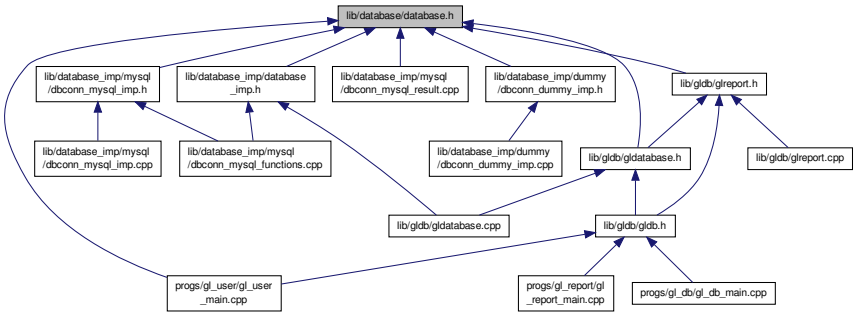
User interface to database functionality.

```
#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:



10.5.1 Detailed Description

User interface to database functionality.

Author

Paul Griffiths

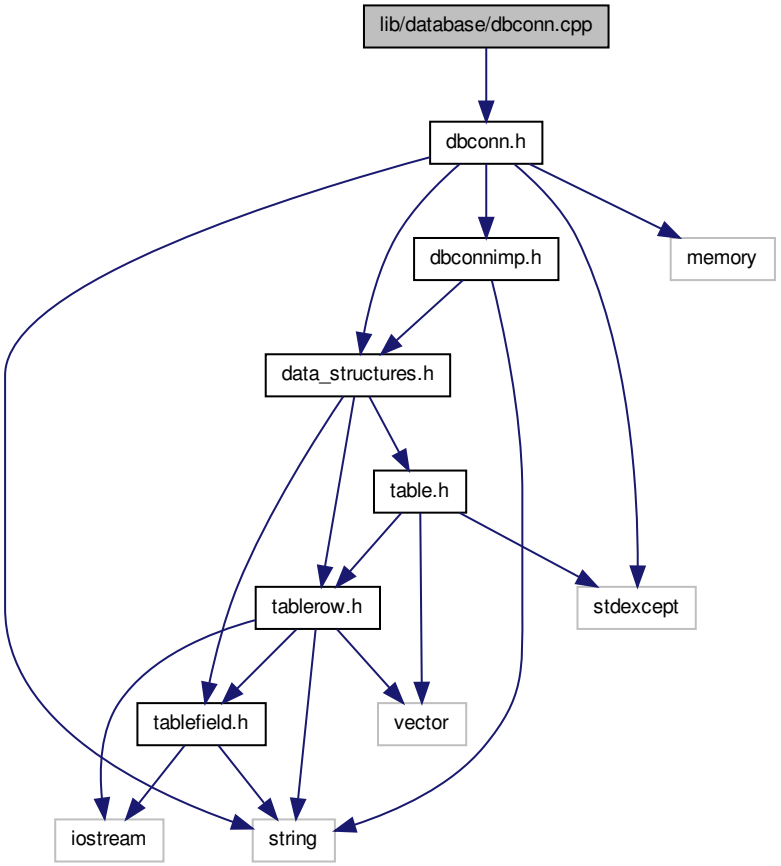
Copyright

10.6 lib/database/dbconn.cpp File Reference

Implementation of database connection class.

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

Include dependency graph for dbconn.cpp:



10.6.1 Detailed Description

Implementation of database connection class.

Author

Paul Griffiths

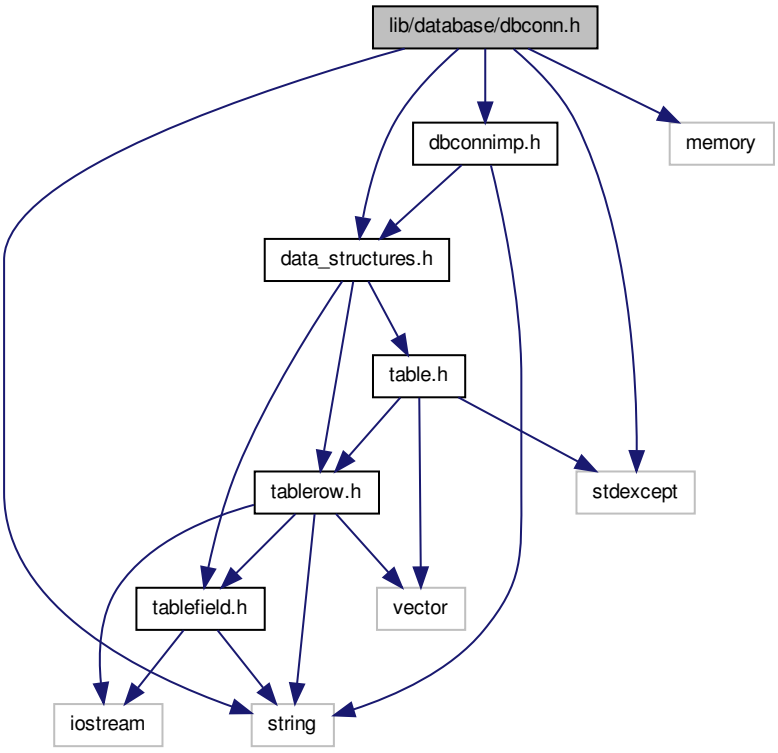
Copyright

10.7 lib/database/dbconn.h File Reference

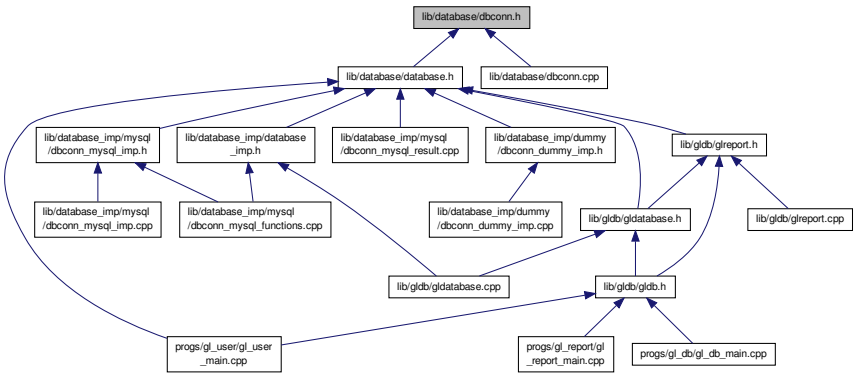
Interface to database connection base class.

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

Include dependency graph for dbconn.h:



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.

10.7.1 Detailed Description

Interface to database connection base class.

Author

Paul Griffiths

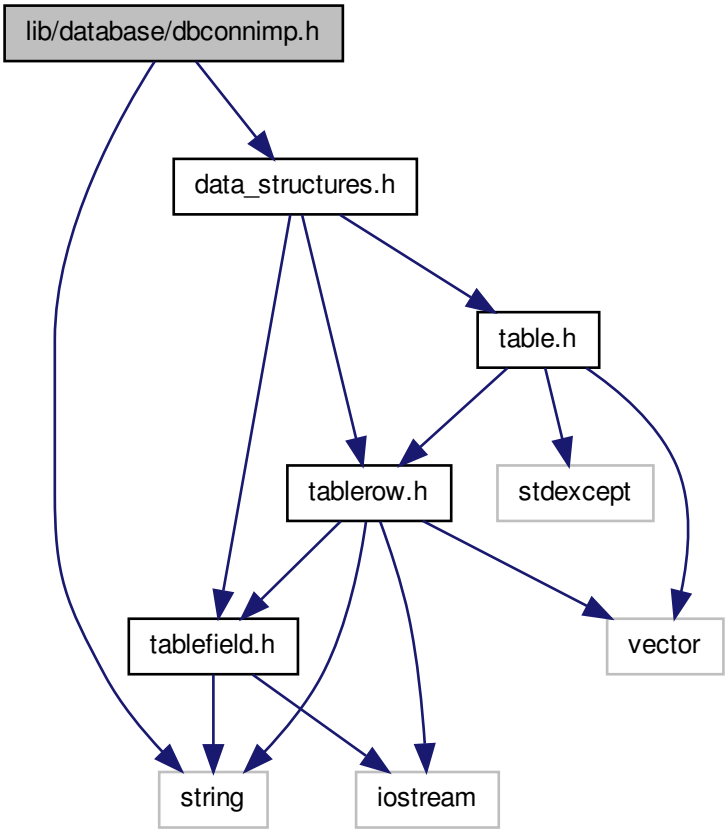
Copyright

10.8 lib/database/dbconnimp.h File Reference

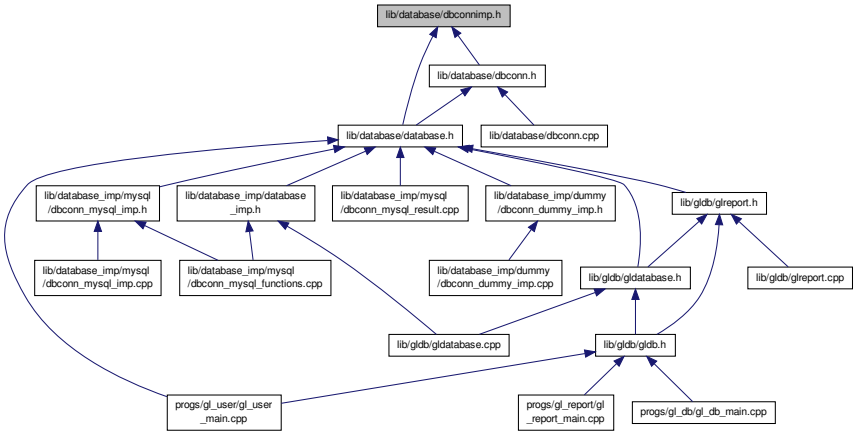
Interface to abstract database implementation base class.

```
#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.

10.8.1 Detailed Description

Interface to abstract database implementation base class.

Author

Paul Griffiths

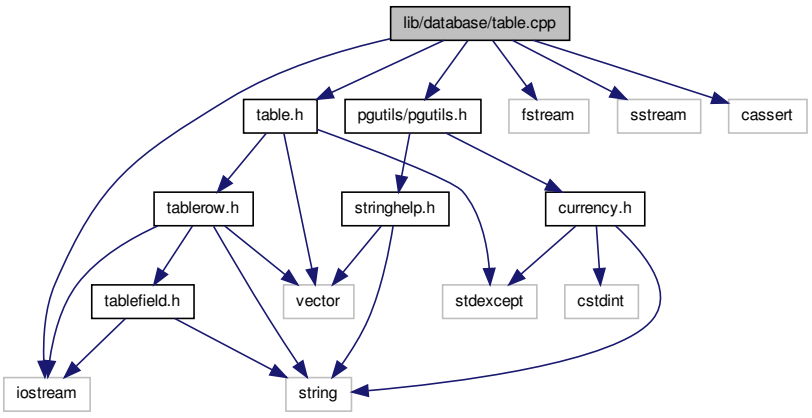
Copyright

10.9 lib/database/table.cpp File Reference

Implementation of database table data structure.

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <cassert>
#include "table.h"
#include "pgutils/pgutils.h"
```

Include dependency graph for table.cpp:



10.9.1 Detailed Description

Implementation of database table data structure.

Author

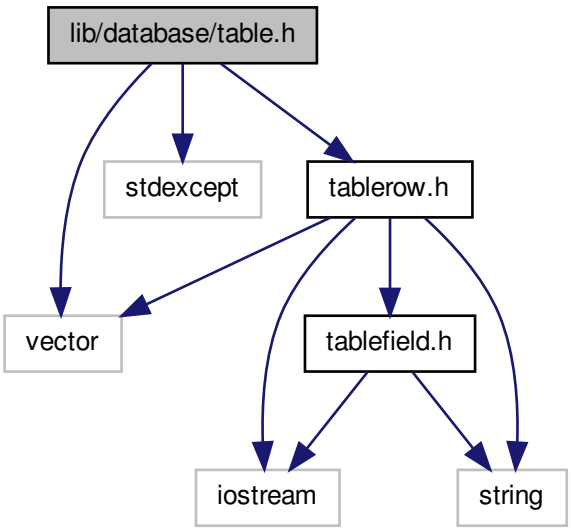
Paul Griffiths

10.10 lib/database/table.h File Reference

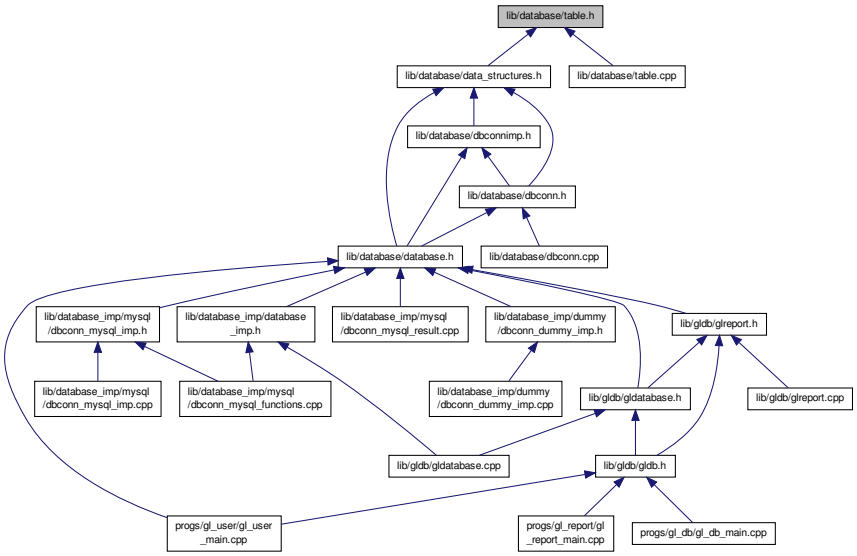
Interface to database table data structure.

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

Include dependency graph for table.h:



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



Classes

- class gldb::TableException

  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.

10.10.1 Detailed Description

Interface to database table data structure.

Author

Paul Griffiths

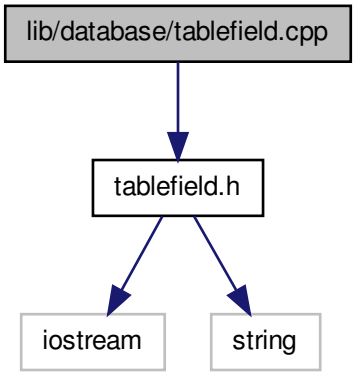
Copyright

10.11 lib/database/tablefield.cpp File Reference

Implementation of database table field class.

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

Include dependency graph for tablefield.cpp:



10.11.1 Detailed Description

Implementation of database table field class.

Author

Paul Griffiths

Copyright

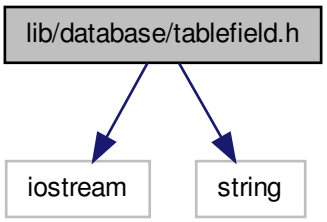
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. http://www.gnu.org/licenses/

10.12 lib/database/tablefield.h File Reference

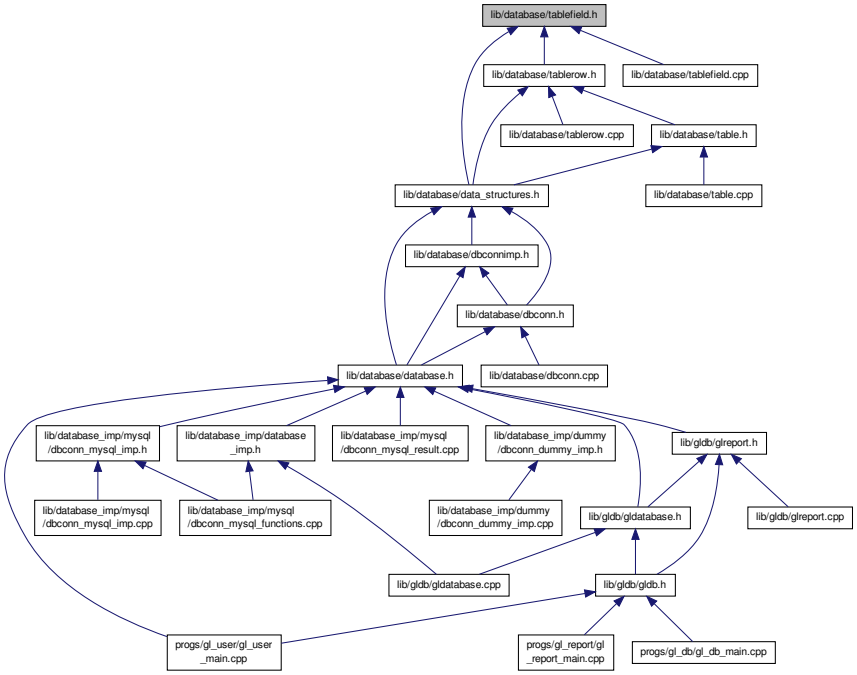
Interface to database table field class.

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

Include dependency graph for tablefield.h:



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



Classes

- class gldb::TableField

  Database table field class.

Functions

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

  Overridden << operator for printing a field.

10.12.1 Detailed Description

Interface to database table field class.

Author

Paul Griffiths

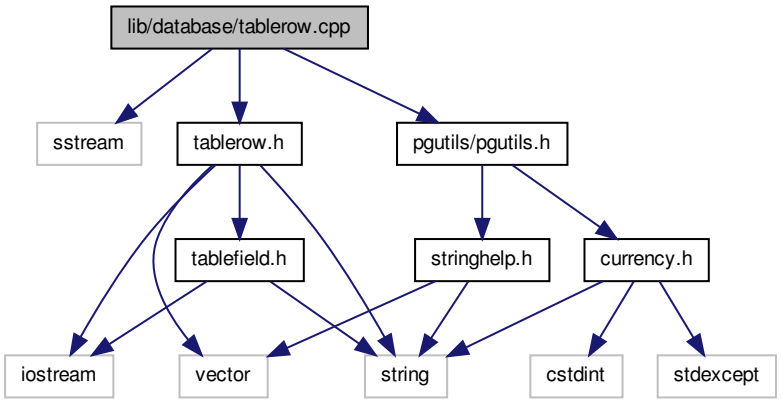
Copyright

10.13 lib/database/tablerow.cpp File Reference

Implementation of database table row data structure.

```
#include <sstream>
#include "tablerow.h"
#include "pgutils/pgutils.h"
```

Include dependency graph for tablerow.cpp:



10.13.1 Detailed Description

Implementation of database table row data structure.

Author

Paul Griffiths

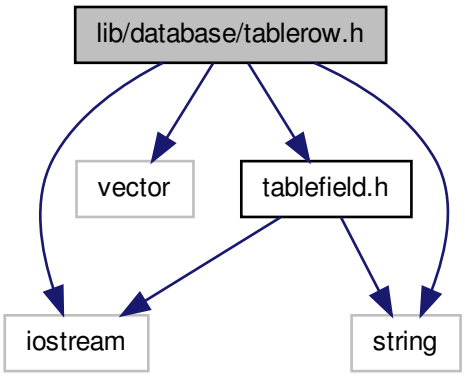
Copyright

10.14 lib/database/tablerow.h File Reference

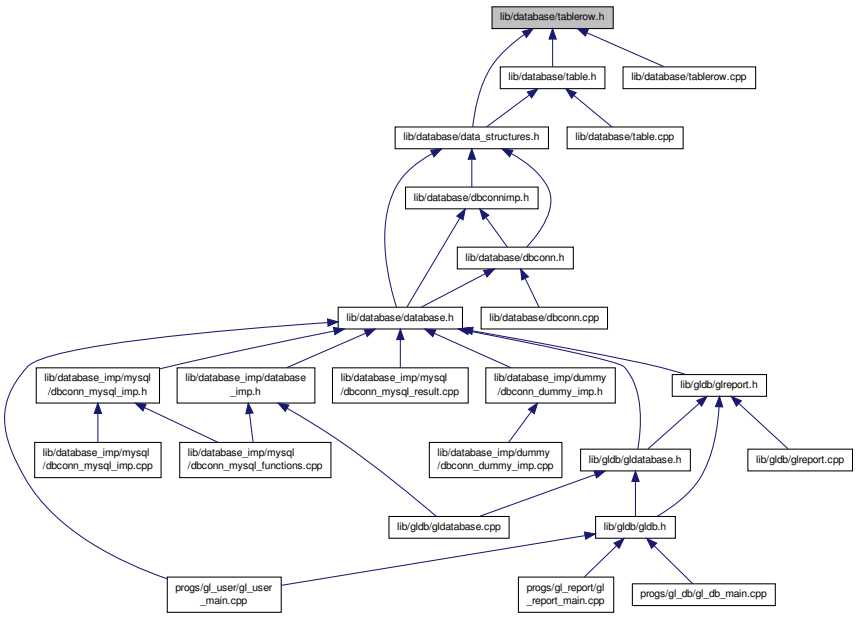
Interface to database table row data structure.

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

Include dependency graph for tablerow.h:



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



Classes

- class gldb::TableRow

  Database table row class.

10.14.1 Detailed Description

Interface to database table row data structure.

Author

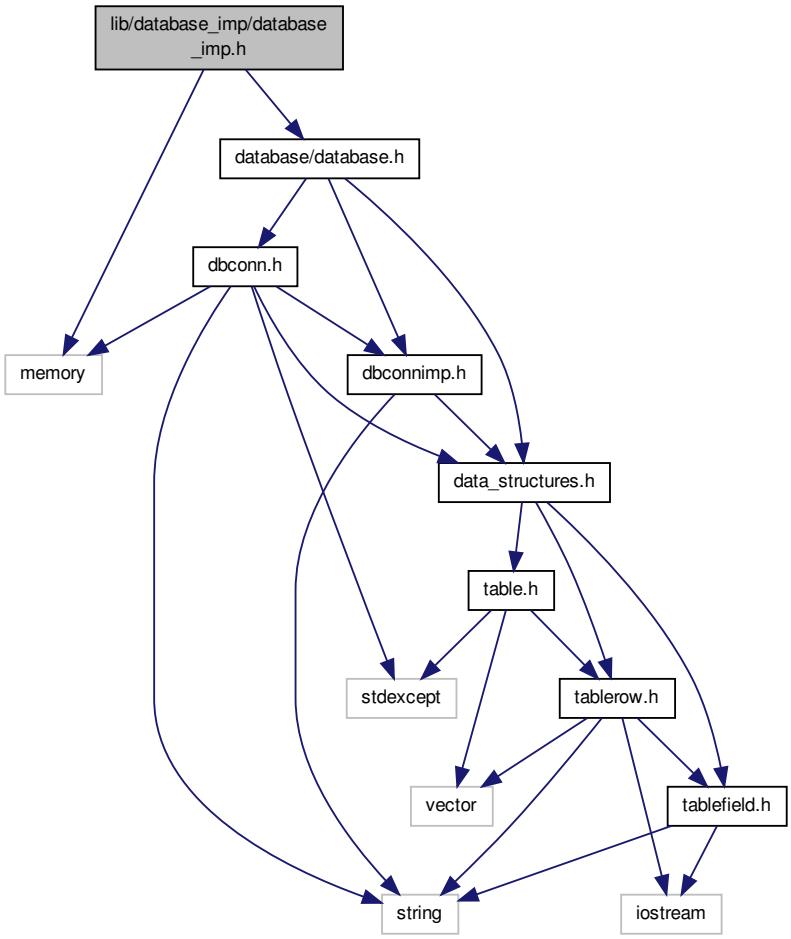
Paul Griffiths

10.15 lib/database_imp/database_imp.h File Reference

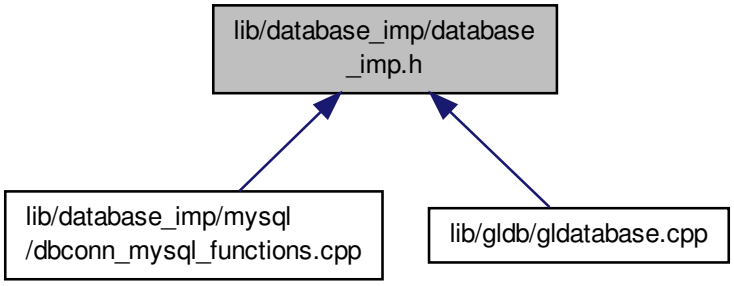
Interface to database implementation factory function.

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

Include dependency graph for database_imp.h:



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



Functions

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

  Creates and returns a pointer to a database implementation.

- std::string gldb::get_database_type ()

  Returns the name of the compiled-in database type.

10.15.1 Detailed Description

Interface to database implementation factory function.

Author

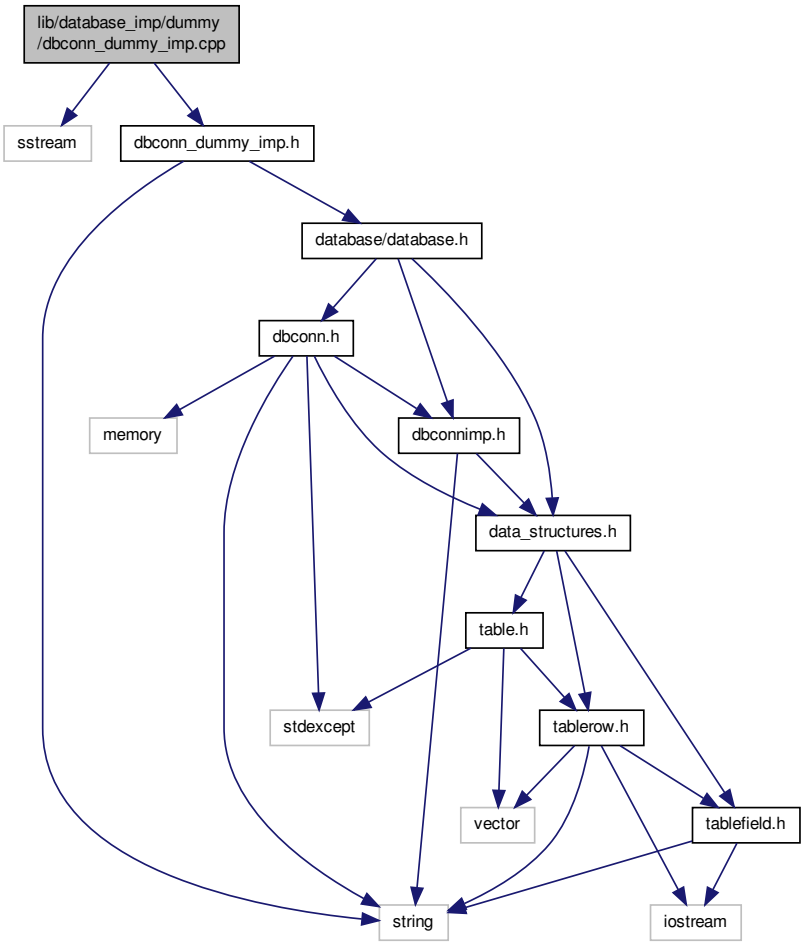
Paul Griffiths

10.16 lib/database_imp/dummy/dbconn_dummy_imp.cpp File Reference

Implementation of Dummy database connection implementation class.

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

Include dependency graph for dbconn_dummy_imp.cpp:



10.16.1 Detailed Description

Implementation of Dummy database connection implementation class.

Author

Paul Griffiths

Copyright

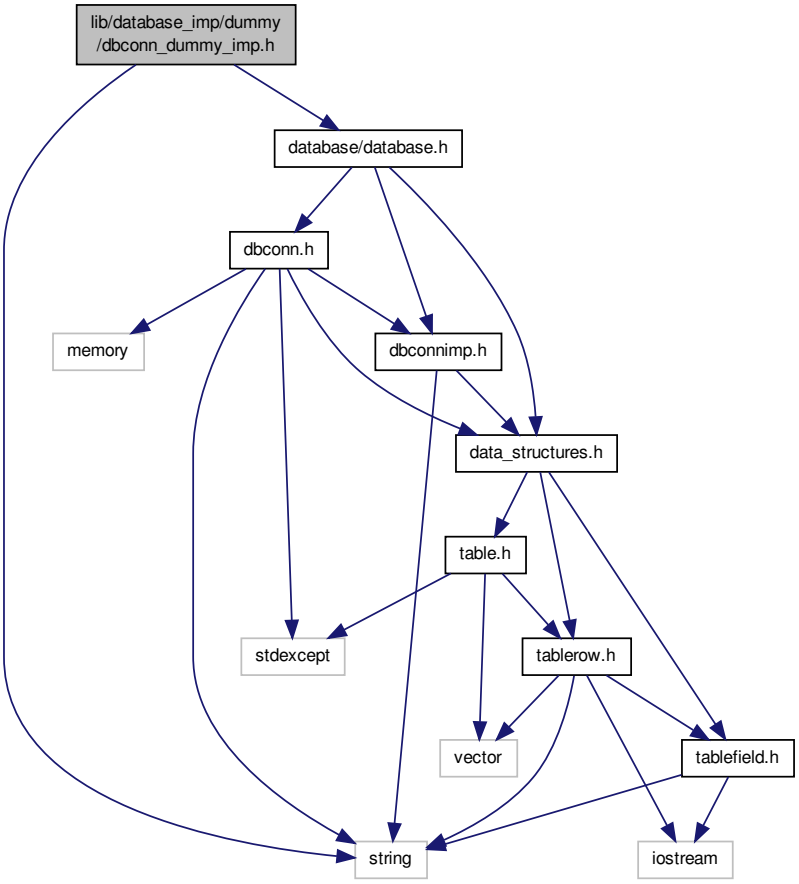
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. http://www.gnu.org/licenses/

10.17 lib/database_imp/dummy/dbconn_dummy_imp.h File Reference

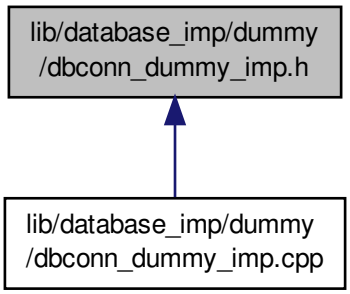
Interface to dummy database connection implementation class.

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

Include dependency graph for dbconn_dummy_imp.h:



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



Classes

- class gldb::DBConnDummy

  Dummy database implementation class.

10.17.1 Detailed Description

Interface to dummy database connection implementation class.

Author

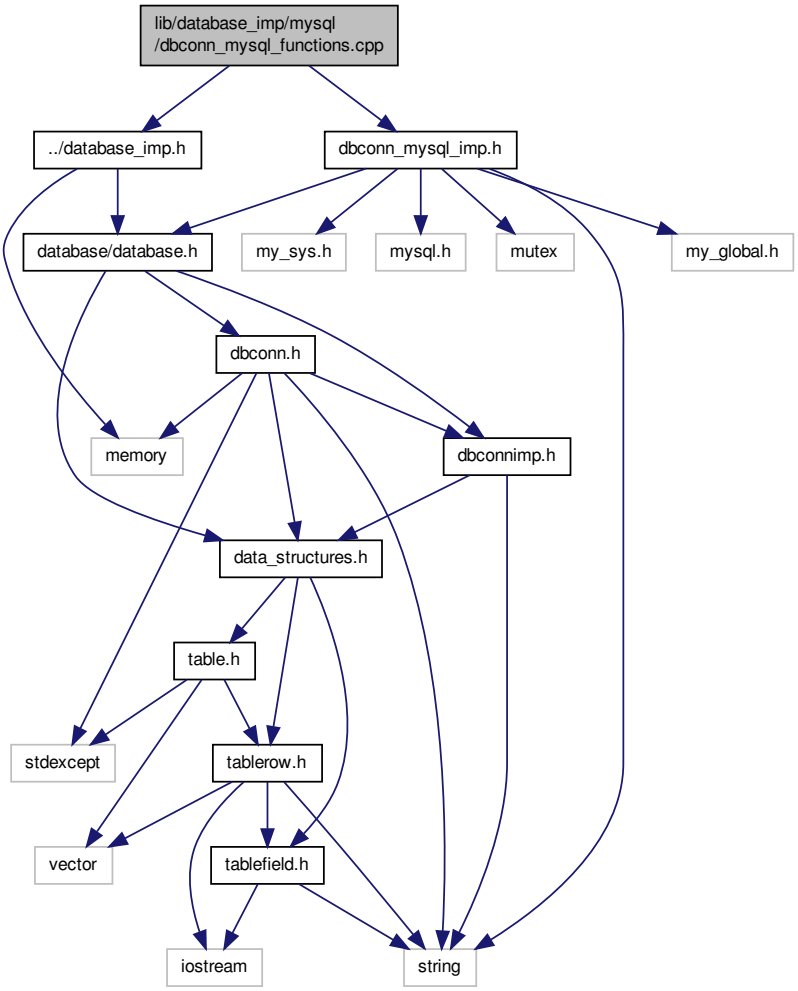
Paul Griffiths

10.18 lib/database_imp/mysql/dbconn_mysql_functions.cpp File Reference

Implementation of MySQL implementation factory function.

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

Include dependency graph for dbconn_mysql_functions.cpp:



10.18.1 Detailed Description

Implementation of MySQL implementation factory function.

Author

Paul Griffiths

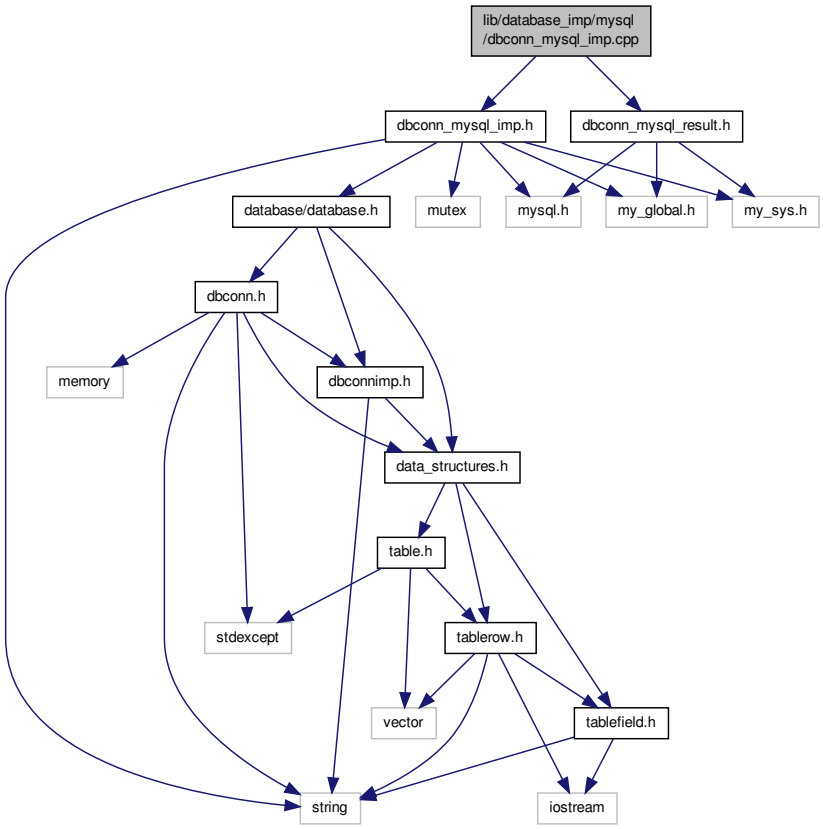
Copyright

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

Implementation of MySQL database connection implementation class.

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

Include dependency graph for dbconn_mysql_imp.cpp:



Functions

- static TableRow get_field_names (MySQLResult &result)

  Gets field names from a MySQL result structure.
- static TableRow get_row (MySQLResult &result, MYSQL_ROW row)

  Creates a TableRow from a MySQL result row.

10.19.1 Detailed Description

Implementation of MySQL database connection implementation class.

Author

Paul Griffiths

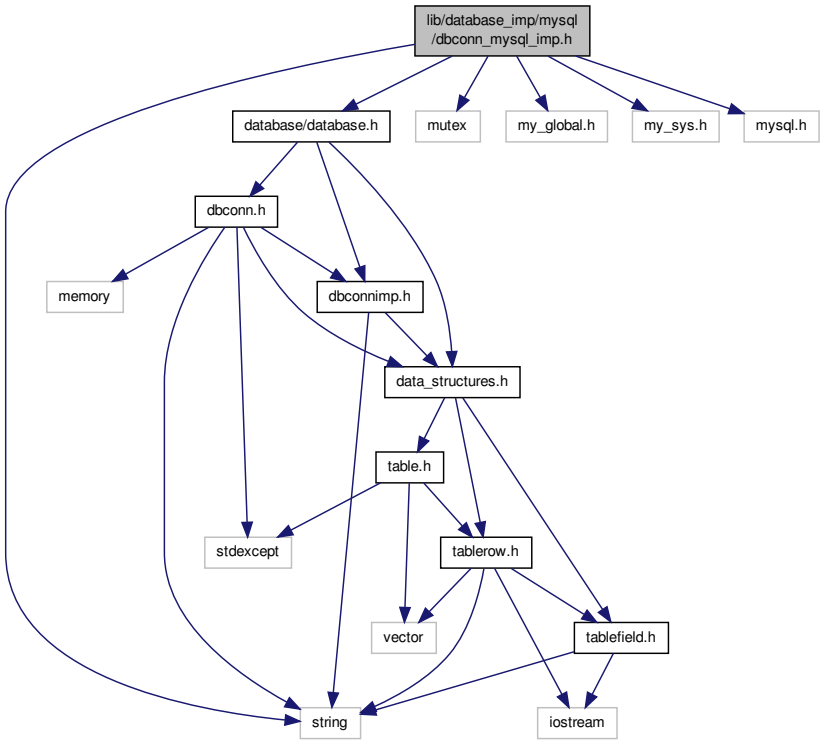
Copyright

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

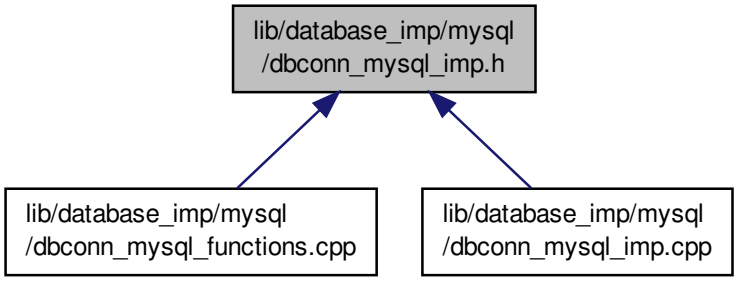
Interface to MySQL database connection implementation class.

```
#include <string>
#include <mutex>
#include "database/database.h"
#include <my_global.h>
#include <my_sys.h>
#include <mysql.h>
```

Include dependency graph for dbconn_mysql_imp.h:



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



Classes

- class gldb::DBConnMySQL

  MySQL database implementation class.

10.20.1 Detailed Description

Interface to MySQL database connection implementation class.

Author

Paul Griffiths

Copyright

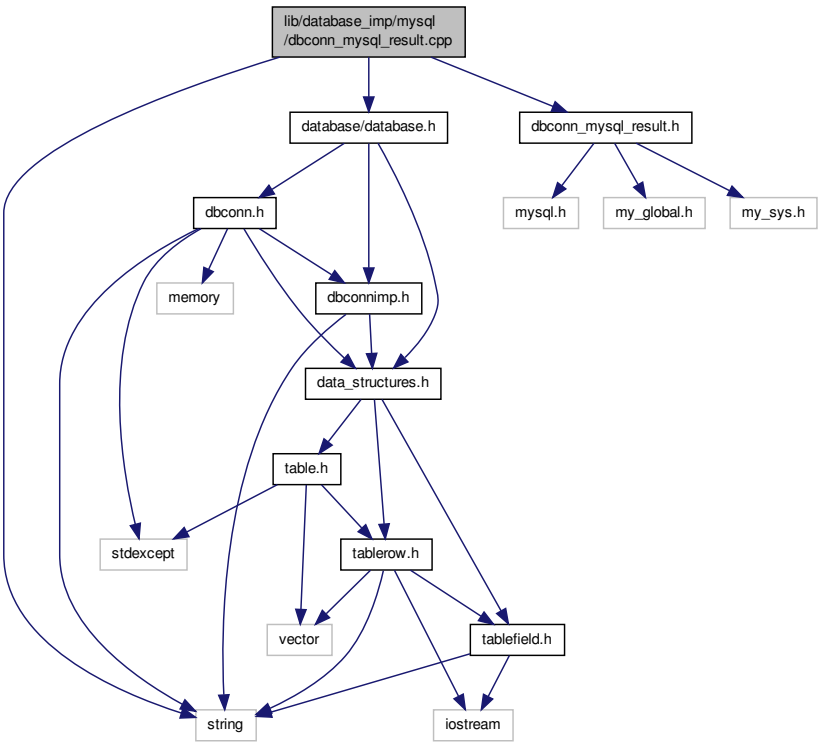
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. http://www.gnu.org/licenses/

10.21 lib/database_imp/mysql/dbconn_mysql_result.cpp File Reference

Implementation of MySQL result structure resource handle class.

```
#include <string>
#include "database/database.h"
#include "dbconn_mysql_result.h"
```

Include dependency graph for dbconn_mysql_result.cpp:



10.21.1 Detailed Description

Implementation of MySQL result structure resource handle class.

Author

Paul Griffiths

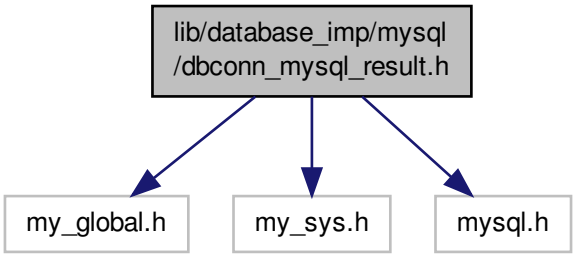
Copyright

10.22 lib/database_imp/mysql/dbconn_mysql_result.h File Reference

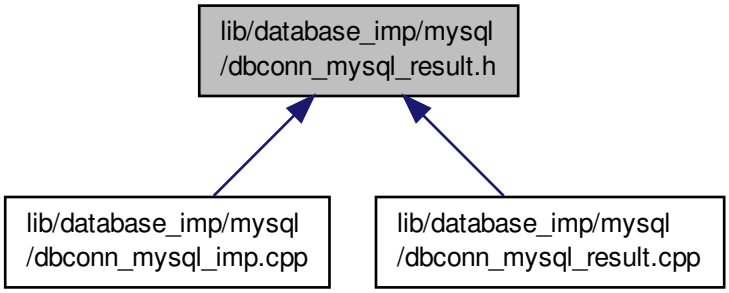
Interface to MySQL result structure resource handle class.

```
#include <my_global.h>
#include <my_sys.h>
#include <mysql.h>
```

Include dependency graph for dbconn_mysql_result.h:



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



Classes

- class gldb::MySQLResult

  MySQL result structure class.

10.22.1 Detailed Description

Interface to MySQL result structure resource handle class.

Author

Paul Griffiths

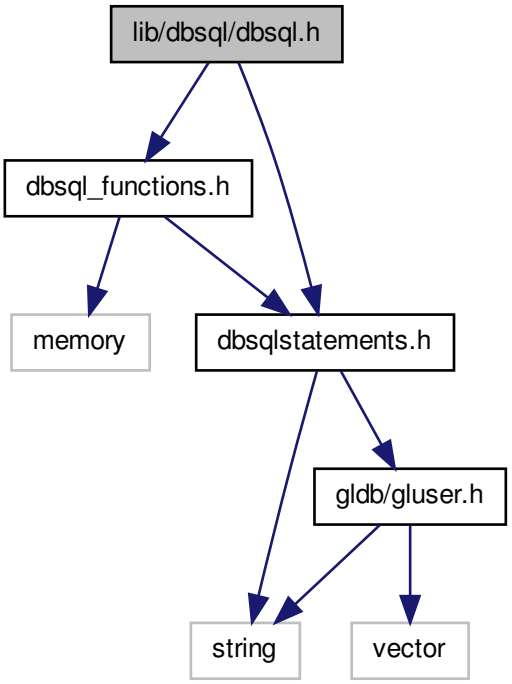
Copyright

10.23 lib/dbsql/dbsql.h File Reference

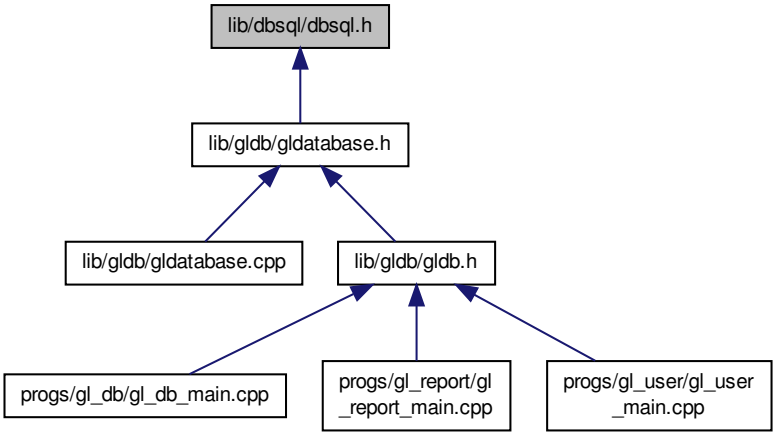
User interface to DBSQL module.

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

Include dependency graph for dbsql.h:



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



10.23.1 Detailed Description

User interface to DBSQL module.

Author

Paul Griffiths

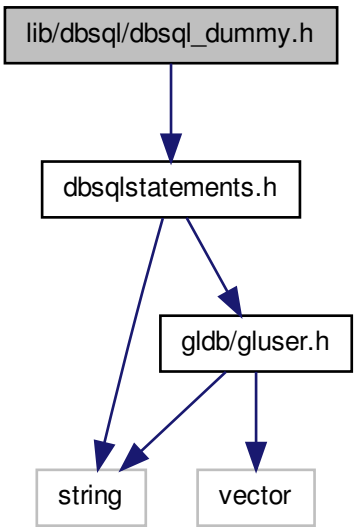
Copyright

10.24 lib/dbsql/dbsql_dummy.h File Reference

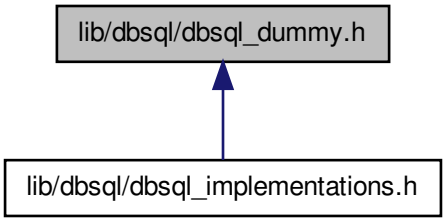
Interface to dummy SQL statement class.

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

Include dependency graph for dbsql_dummy.h:



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



Classes

- class genleg::DBSQLDummy

  Dummy SQL statements class.

10.24.1 Detailed Description

Interface to dummy SQL statement class.

Author

Paul Griffiths

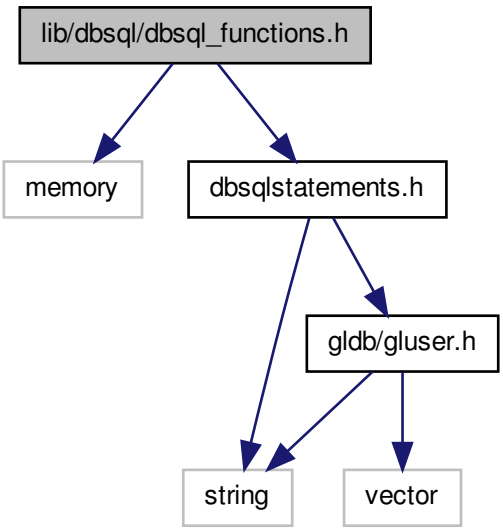
Copyright

10.25 lib/dbsql/dbsql_functions.h File Reference

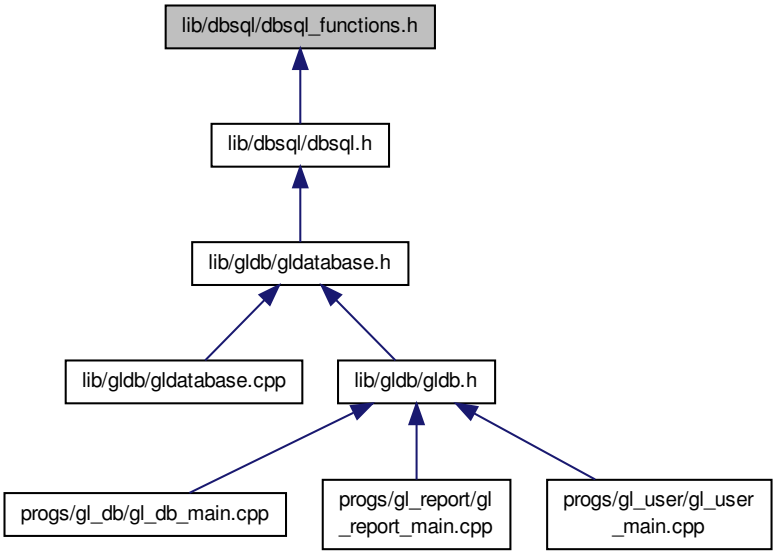
Interface to SQL module standalone functions.

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

Include dependency graph for dbsql_functions.h:



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



Functions

- std::shared_ptr< const DBSQLStatements > **genleg::get_sql_object** ()

  Factory function for DBSQL objects.

10.25.1 Detailed Description

Interface to SQL module standalone functions.

Author

Paul Griffiths

Copyright

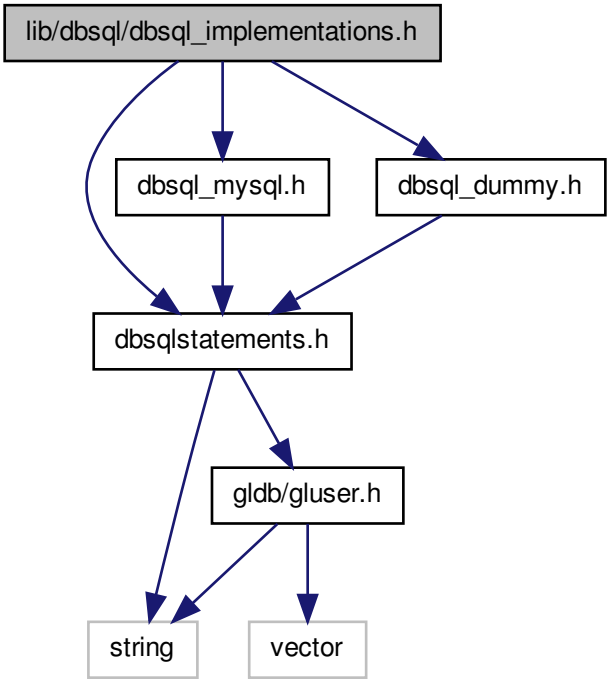
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. http://www.gnu.org/licenses/

10.26 lib/dbsql/dbsql_implementations.h File Reference

Aggregation header for DBSqlStatements implementations.

```
#include "dbsqlstatements.h"
#include "dbsql_mysql.h"
#include "dbsql_dummy.h"
```

Include dependency graph for dbsql_implementations.h:



10.26.1 Detailed Description

Aggregation header for DBSqlStatements implementations.

Author

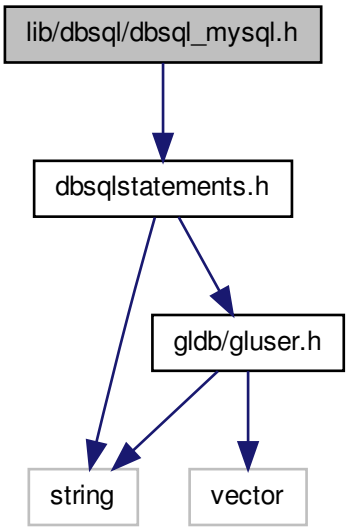
Paul Griffiths

10.27 lib/dbsql/dbsql_mysql.h File Reference

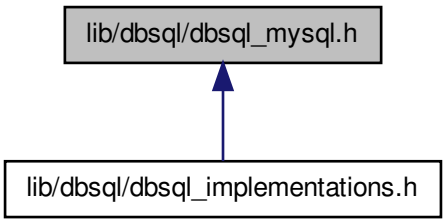
Interface to MySQL SQL statement class.

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

Include dependency graph for dbsql_mysql.h:



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



Classes

- class genleg::DBSQLMySQL

MySQL SQL statements class.

10.27.1 Detailed Description

Interface to MySQL SQL statement class.

Author

Paul Griffiths

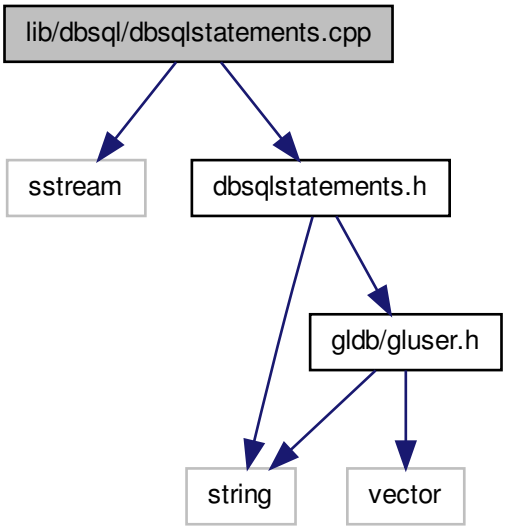
Copyright

10.28 lib/dbsql/dbsqlstatements.cpp File Reference

Implementation of SQL statement class.

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

Include dependency graph for dbsqlstatements.cpp:



10.28.1 Detailed Description

Implementation of SQL statement class.

Author

Paul Griffiths

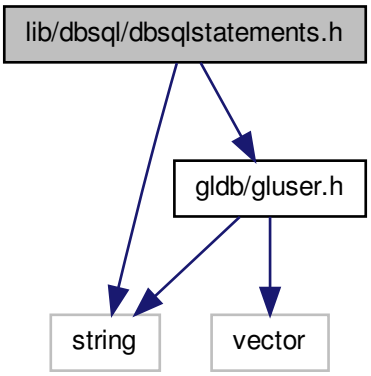
Copyright

10.29 lib/dbsql/dbsqlstatements.h File Reference

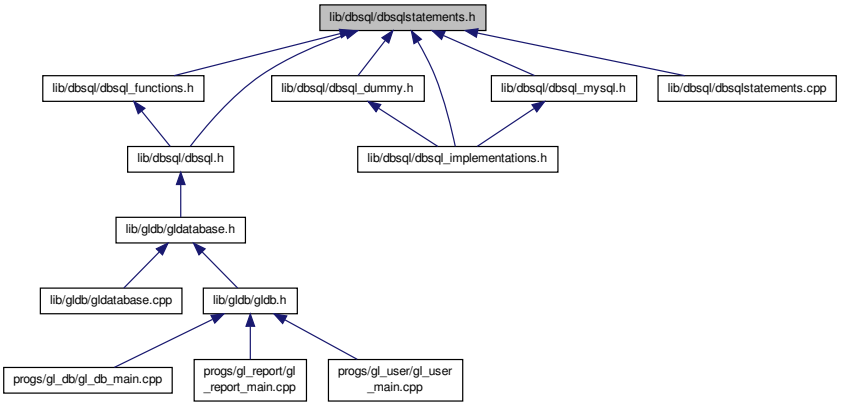
Implementation of SQL module standalone functions.

```
#include <string>
#include "gldb/gluser.h"
```

Include dependency graph for dbsqlstatements.h:



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



Classes

- class genleg::DBSQLStatements

  SQL statements class.

10.29.1 Detailed Description

Implementation of SQL module standalone functions. Interface to SQL statements class.

Author

Paul Griffiths

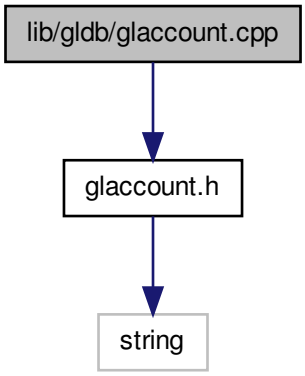
Copyright

10.30 lib/gldb/glaccount.cpp File Reference

Implementation of nominal account class.

```
#include "glaccount.h"
```

Include dependency graph for glaccount.cpp:



10.30.1 Detailed Description

Implementation of nominal account class.

Author

Paul Griffiths

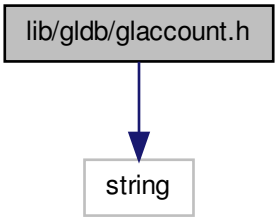
Copyright

10.31 lib/gldb/glaccount.h File Reference

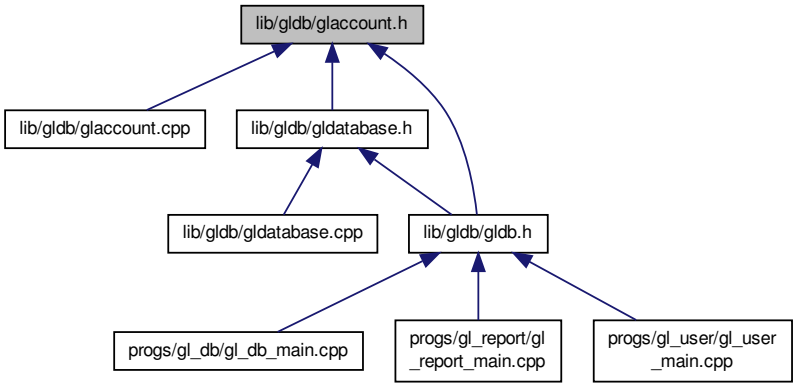
Interface to nominal account class.

```
#include <string>
```

Include dependency graph for glaccount.h:



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



Classes

- class genleg::GLAccount

  Nominal account class.

10.31.1 Detailed Description

Interface to nominal account class.

Author

Paul Griffiths

Copyright

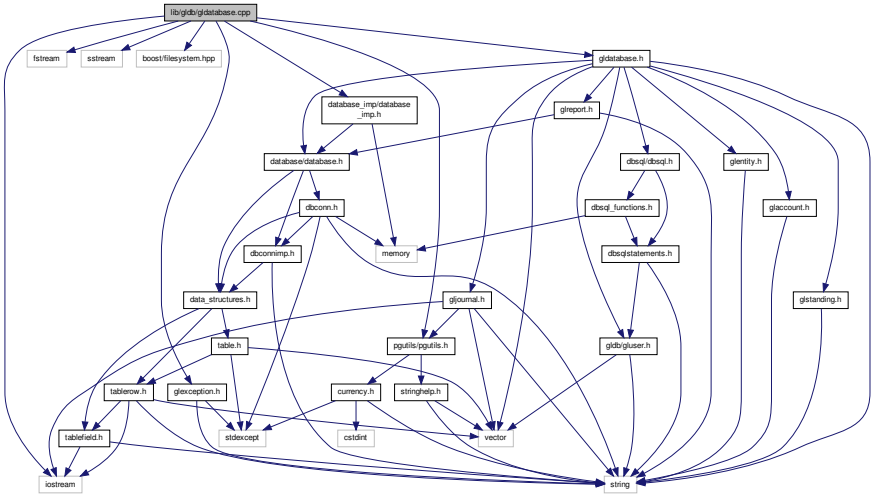
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. http://www.gnu.org/licenses/

10.32 lib/gldb/gldatabase.cpp File Reference

Implementation of General Ledger database class.

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <boost/filesystem.hpp>
#include "gldatabase.h"
#include "glexception.h"
#include "database_imp/database_imp.h"
#include "pgutils/pgutils.h"
```

Include dependency graph for gldatabase.cpp:



Functions

- static bool boolstring_to_bool (const std::string &bs)

  Converts a string representation of a bool to a bool.
- **m_views** ({"current_trial_balance","check_total","all_jes"})

10.32.1 Detailed Description

Implementation of General Ledger database class.

Author

Paul Griffiths

Copyright

10.32.2 Function Documentation

10.32.2.1 static bool boolstring_to_bool (const std::string & *bs*) [static]

Converts a string representation of a bool to a bool.

Parameters

bs	The bool string.

Returns

`true` if `bs` contains "1" or "TRUE", `false` if `bs` contains "0" or "FALSE".

Exceptions

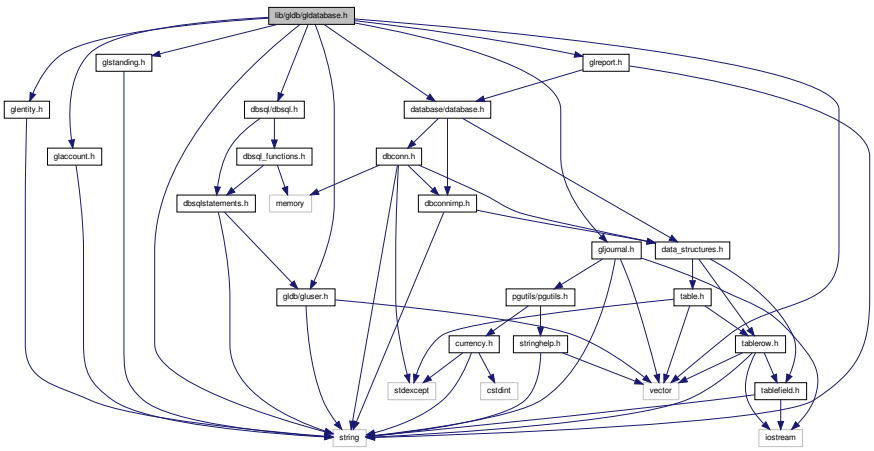
GLDBException	if `bs` contains any other value.

10.33 lib/gldb/gldatabase.h File Reference

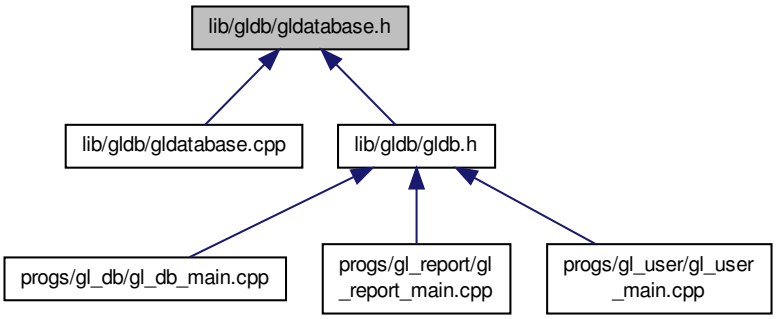
Interface to General Ledger database class.

```
#include <vector>
#include <string>
#include "database/database.h"
#include "dbsql/dbsql.h"
#include "gluser.h"
#include "glreport.h"
#include "gljournal.h"
#include "glentity.h"
#include "glaccount.h"
#include "glstanding.h"
```

Include dependency graph for gldatabase.h:



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



Classes

- class genleg::GLDatabase

  General ledger database class.
- class genleg::GLDBTransaction

  Database transaction RAII class.

10.33.1 Detailed Description

Interface to General Ledger database class.

Author

Paul Griffiths

Copyright

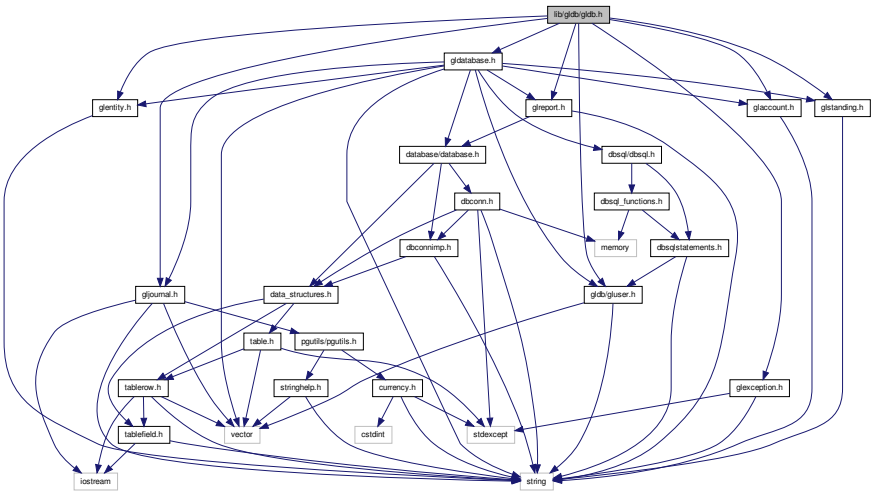
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. http://www.gnu.org/licenses/

10.34 lib/gldb/gldb.h File Reference

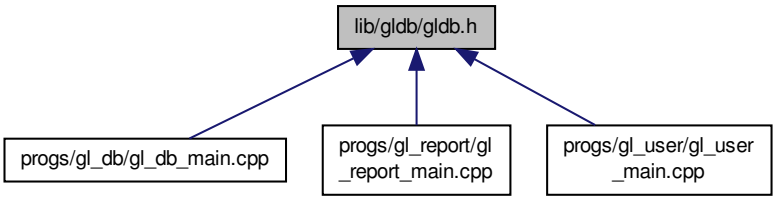
User interface to General Ledger database module.

```
#include "glexception.h"
#include "gldatabase.h"
#include "gluser.h"
#include "glreport.h"
#include "gljournal.h"
#include "glentity.h"
#include "glaccount.h"
#include "glstanding.h"
```

Include dependency graph for gldb.h:



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



10.34.1 Detailed Description

User interface to General Ledger database module.

Author

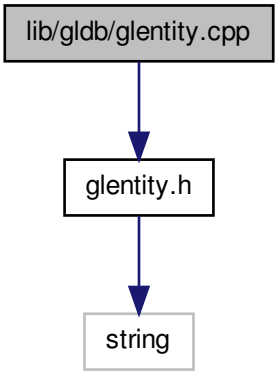
Paul Griffiths

Copyright

10.35 lib/gldb/glentity.cpp File Reference

Implementation of general ledger entity class.

```
#include "glentity.h"
```
Include dependency graph for glentity.cpp:



10.35.1 Detailed Description

Implementation of general ledger entity class.

Author

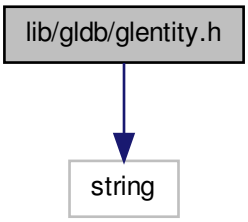
Paul Griffiths

Copyright

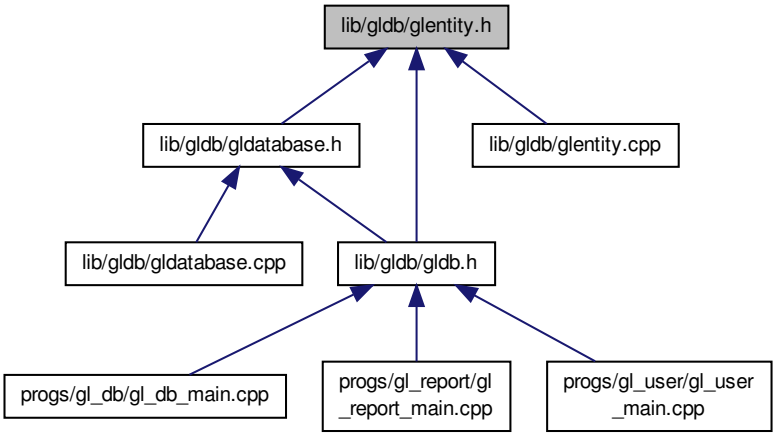
10.36 lib/gldb/glentity.h File Reference

Interface to general ledger entity class.

```
#include <string>
```
Include dependency graph for glentity.h:



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



Classes

- class genleg::GLEntity

  General ledger entity class.

10.36.1 Detailed Description

Interface to general ledger entity class.

Author

Paul Griffiths

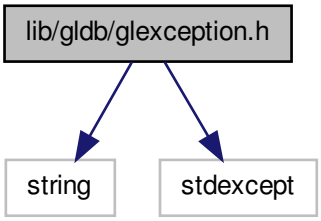
Copyright

10.37 lib/gldb/glexception.h File Reference

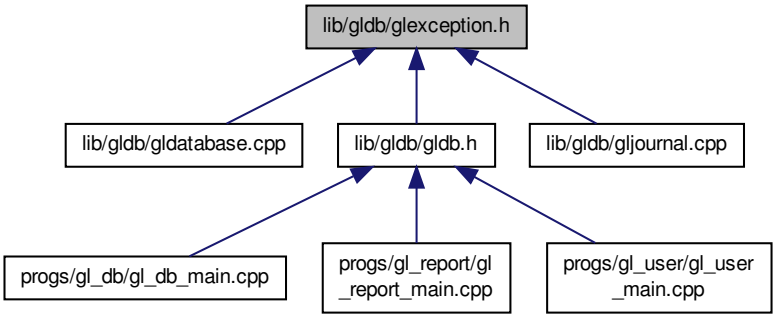
Interface to General Ledger base exception class.

```
#include <string>
#include <stdexcept>
```

Include dependency graph for glexception.h:



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



Classes

- class genleg::GLDBException

  Base general ledger database exceptionc class.

10.37.1 Detailed Description

Interface to General Ledger base exception class.

Author

Paul Griffiths

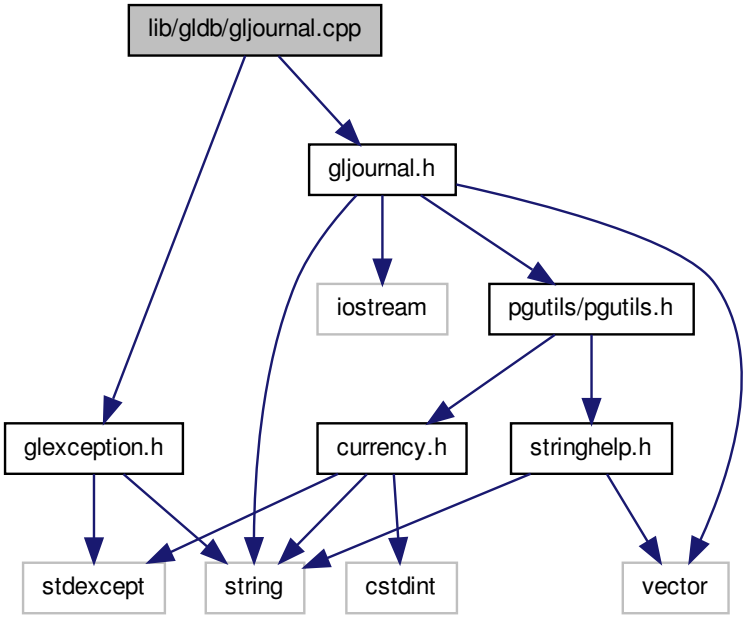
Copyright

10.38 lib/gldb/gljournal.cpp File Reference

Implementation of journal entry classes.

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

Include dependency graph for gljournal.cpp:



10.38.1 Detailed Description

Implementation of journal entry classes.

Author

Paul Griffiths

Copyright

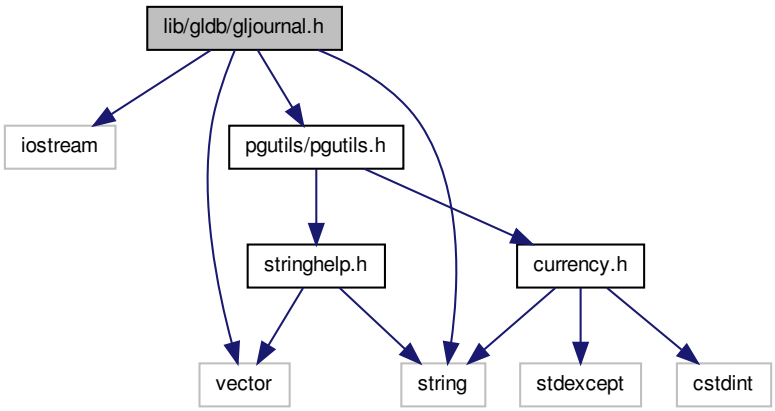
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. http://www.gnu.org/licenses/

10.39 lib/gldb/gljournal.h File Reference

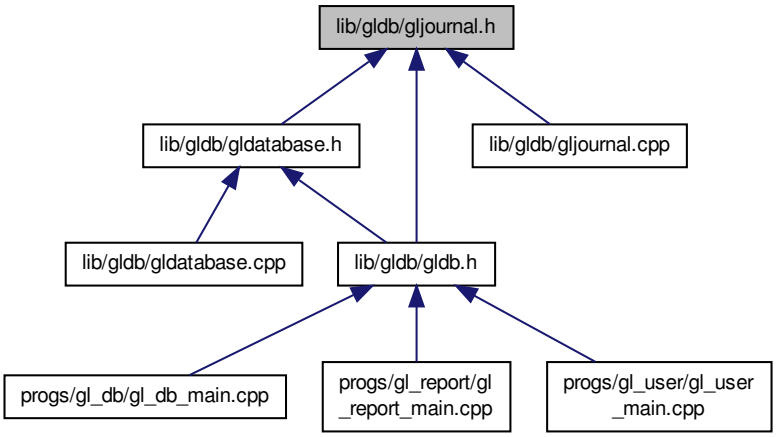
Interface to journal entry classes.

```
#include <iostream>
#include <vector>
#include <string>
#include "pgutils/pgutils.h"
```

Include dependency graph for gljournal.h:



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



Classes

- class genleg::GLJELine

  Journal entry line class.
- class genleg::GLJournal

  Journal entry class.

Functions

- GLJournal **genleg::journal_from_stream** (std::istream &ifs)

  Returns a journal entry from a stream in a standard format.

10.39.1 Detailed Description

Interface to journal entry classes.

Author

Paul Griffiths

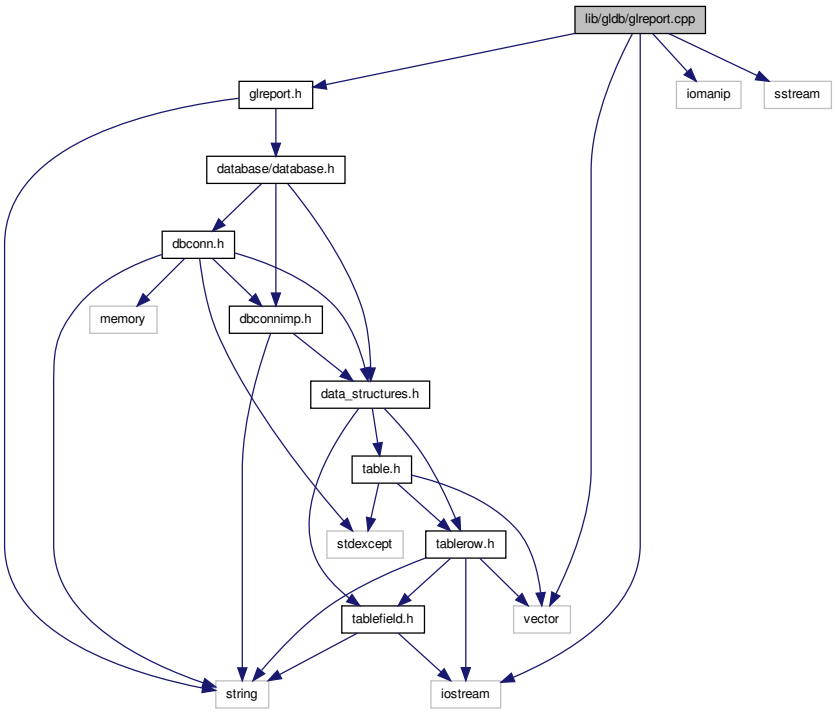
Copyright

10.40 lib/gldb/glreport.cpp File Reference

Implementation of report class.

```
#include <vector>
#include <iomanip>
#include <iostream>
#include <sstream>
#include "glreport.h"
```

Include dependency graph for glreport.cpp:



Functions

- static std::vector< size_t > max_column_widths (const gldb::Table &table)

  Calculates the maximum required column widths for a table.
- static void grow_widths (std::vector< size_t > &widths, const TableRow &row)

Increments a vector of required column widths.

- static std::string separator_row (const std::vector< size_t > &widths)

  Returns a decorated separator row for a table.

- static std::string plain_row (const TableRow &row, const std::vector< size_t > &widths)

  Returns a row for a plain report.

- static std::string decorated_row (const TableRow &row, const std::vector< size_t > &widths)

  Returns a row for a decorated report.

10.40.1 Detailed Description

Implementation of report class.

Author

Paul Griffiths

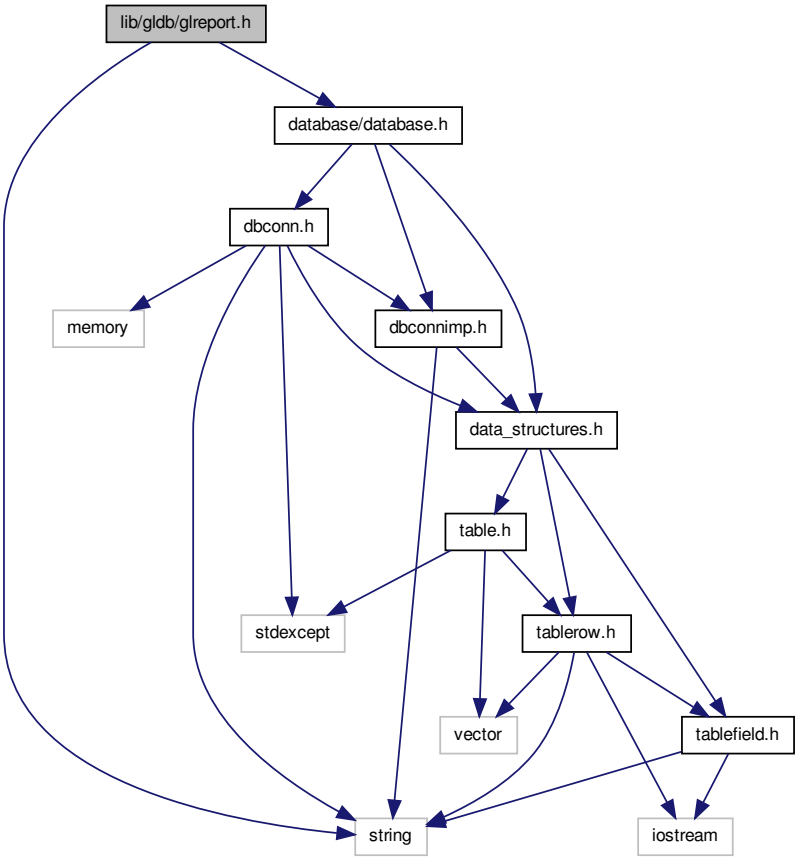
Copyright

10.41 lib/gldb/glreport.h File Reference

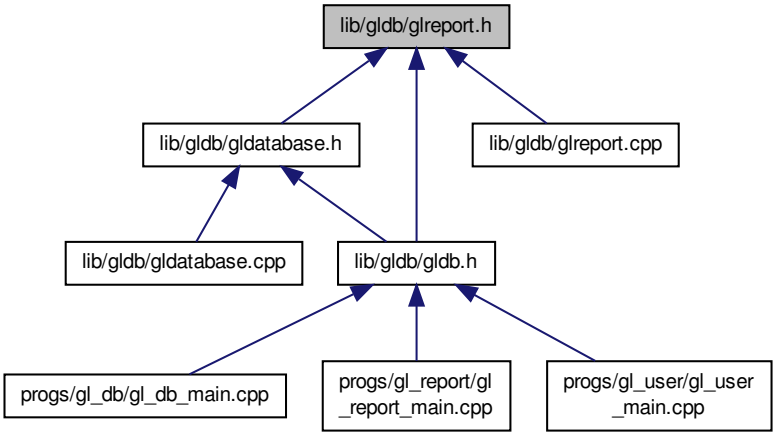
Interface to report class.

```
#include <string>
#include <database/database.h>
```

Include dependency graph for glreport.h:



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



Classes

- class genleg::GLReport

  General ledger report class.

Functions

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

  Creates a plain report from a table.
- std::string genleg::decorated_report_from_table (const gldb::Table &table)

  Creates a decorated report from a table.
- std::ostream & **genleg::operator**<< (std::ostream &out, const GLReport &report)

  Overridden << operator for printing a report.

10.41.1 Detailed Description

Interface to report class.

Author

Paul Griffiths

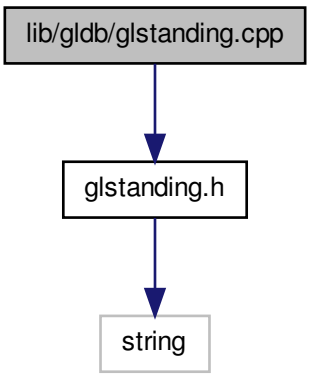
Copyright

10.42 lib/gldb/glstanding.cpp File Reference

Implementation of general ledger standing data class.

```
#include "glstanding.h"
```

Include dependency graph for glstanding.cpp:



10.42.1 Detailed Description

Implementation of general ledger standing data class.

Author

Paul Griffiths

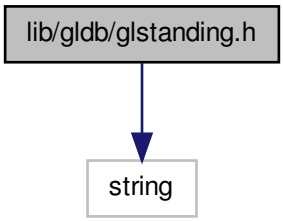
Copyright

10.43 lib/gldb/glstanding.h File Reference

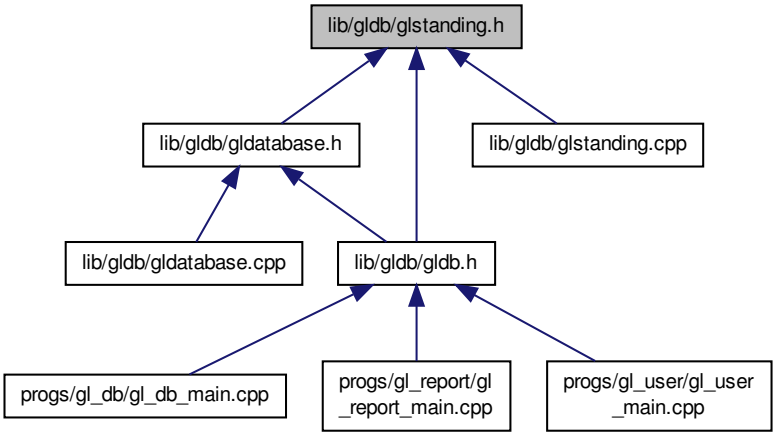
Interface to general ledger standing data class.

```
#include <string>
```

Include dependency graph for glstanding.h:



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



Classes

- class genleg::GLStandingData
  General ledger standing data class.

10.43.1 Detailed Description

Interface to general ledger standing data class.

Author

Paul Griffiths

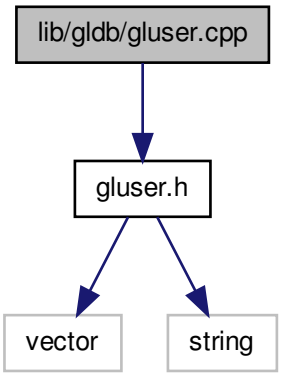
Copyright

10.44 lib/gldb/gluser.cpp File Reference

Implementation of user class.

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

Include dependency graph for gluser.cpp:



10.44.1 Detailed Description

Implementation of user class.

Author

Paul Griffiths

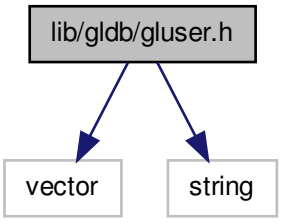
Copyright

10.45 lib/gldb/gluser.h File Reference

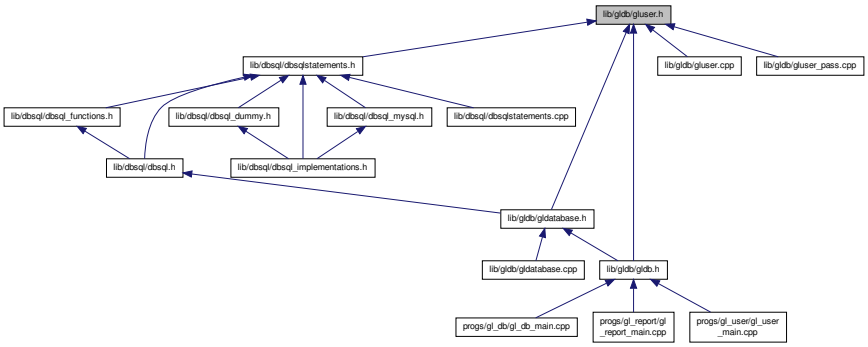
Interface to user class.

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

Include dependency graph for gluser.h:



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



Classes

- class genleg::GLUser

  General ledger user class.

10.45.1 Detailed Description

Interface to user class.

Author

Paul Griffiths

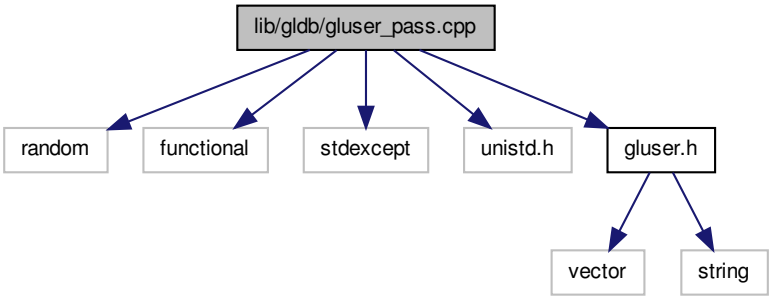
Copyright

10.46 lib/gldb/gluser_pass.cpp File Reference

Implementation of password functions for user class.

```
#include <random>
#include <functional>
#include <stdexcept>
#include <unistd.h>
#include "gluser.h"
```

Include dependency graph for gluser_pass.cpp:



Macros

- #define _XOPEN_SOURCE 600

Functions

- static std::string generate_salt ()

  Generates a random two-character salt for crypt()

10.46.1 Detailed Description

Implementation of password functions for user class.

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

Author

Paul Griffiths

Copyright

10.46.2 Macro Definition Documentation

10.46.2.1 #define _XOPEN_SOURCE 600

UNIX feature test macro

10.46.3 Function Documentation

10.46.3.1 static std::string generate_salt ( ) [static]

Generates a random two-character salt for crypt()

Returns

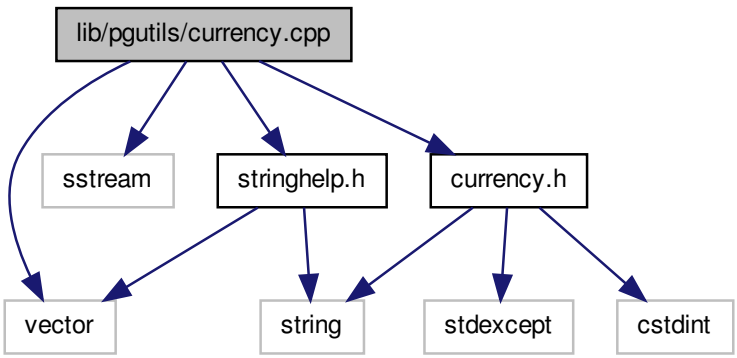
The two-character salt.

10.47 lib/pgutils/currency.cpp File Reference

Implementation of currency amount class.

```
#include <vector>
#include <sstream>
#include "currency.h"
#include "stringhelp.h"
```

Include dependency graph for currency.cpp:



10.47.1 Detailed Description

Implementation of currency amount class.

Author

Paul Griffiths

Copyright

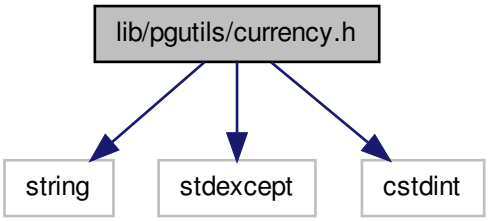
Copyright 2014 Paul Griffiths. Distributed under the terms of the GNU General Public License. http://www.gnu.org/licenses/

10.48 lib/pgutils/currency.h File Reference

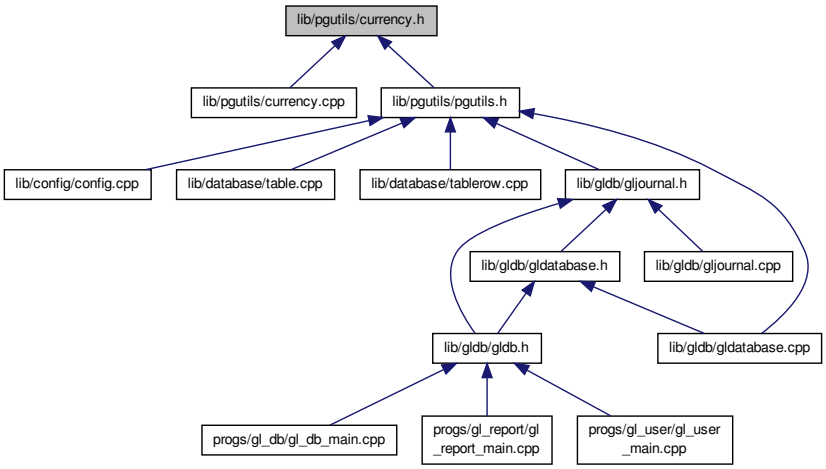
Interface to currency amount class.

```
#include <string>
#include <stdexcept>
#include <cstdint>
```

Include dependency graph for currency.h:



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



Classes

- class pgutils::CurrencyException

  Base Currency exception class.
- class pgutils::Currency

  Currency amount class.

Functions

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

Currency addition operator.

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

  Currency subtraction operator.

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

  Currency equality comparison operator.

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

  Currency inequality comparison operator.

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

  Currency less than comparison operator.

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

  Currency greater than comparison operator.

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

  Currency less than or equal to comparison operator.

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

  Currency greater than or equal to comparison operator.

- Currency pgutils::currency_from_string (const std::string &s)

  Creates a currency amount from a string representation.

10.48.1 Detailed Description

Interface to currency amount class.

Author

Paul Griffiths

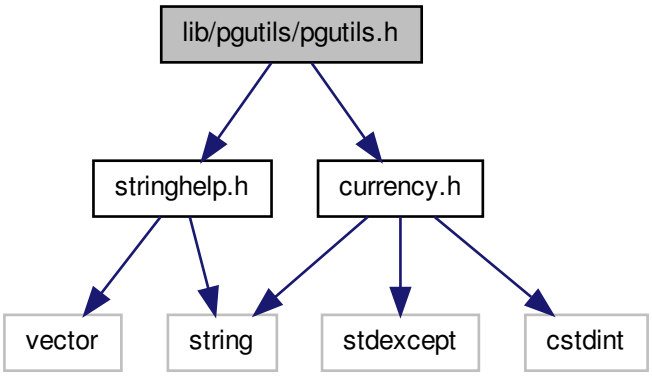
Copyright

10.49 lib/pgutils/pgutils.h File Reference

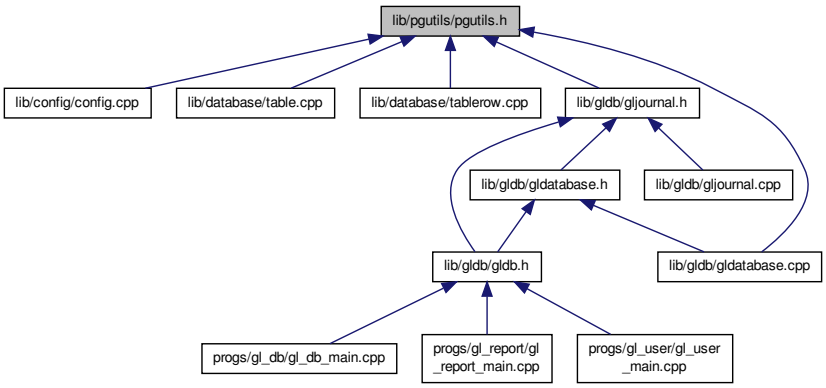
Aggregate interface to general utility functions.

```
#include "stringhelp.h"
#include "currency.h"
```

Include dependency graph for pgutils.h:



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



10.49.1 Detailed Description

Aggregate interface to general utility functions.

Author

Paul Griffiths

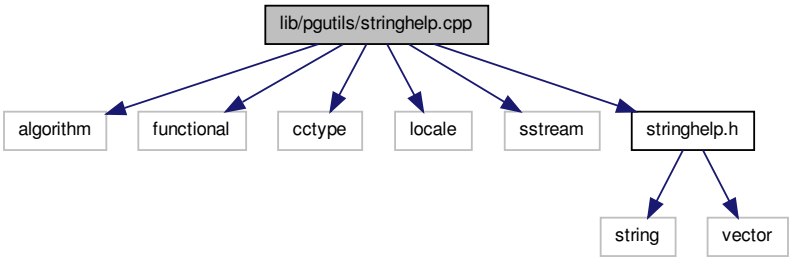
Copyright

10.50 lib/pgutils/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:



10.50.1 Detailed Description

Implementation of string helper functions.

Author

Paul Griffiths

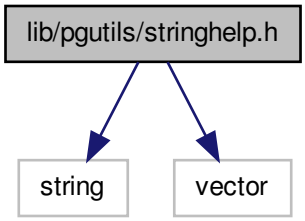
Copyright

10.51 lib/pgutils/stringhelp.h File Reference

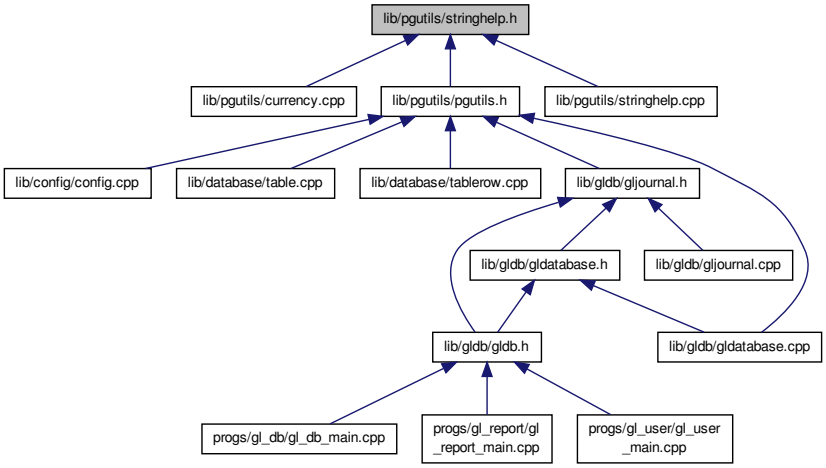
Interface to string helper functions.

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

Include dependency graph for stringhelp.h:



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



Functions

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

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

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

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

  Splits a delimited string into tokens.
- std::vector< std::string > & pgutils::split (std::vector< std::string > &vec, const std::string &s, const char delim)

  Splits a delimited string into tokens.
- bool pgutils::next_content_line (std::istream &ifs, std::string &s)

  Gets the next content line from a stream.
- std::vector< std::string > & pgutils::content_lines (std::vector< std::string > &vec, std::istream &ifs)

Populates a vector of content lines from a stream.

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

  Populates a vector of vectors of fields from a stream.

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

  Joins a vector of strings into a delimited line.

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

  Replaces a substring with another string.

10.51.1 Detailed Description

Interface to string helper functions.

Author

Paul Griffiths

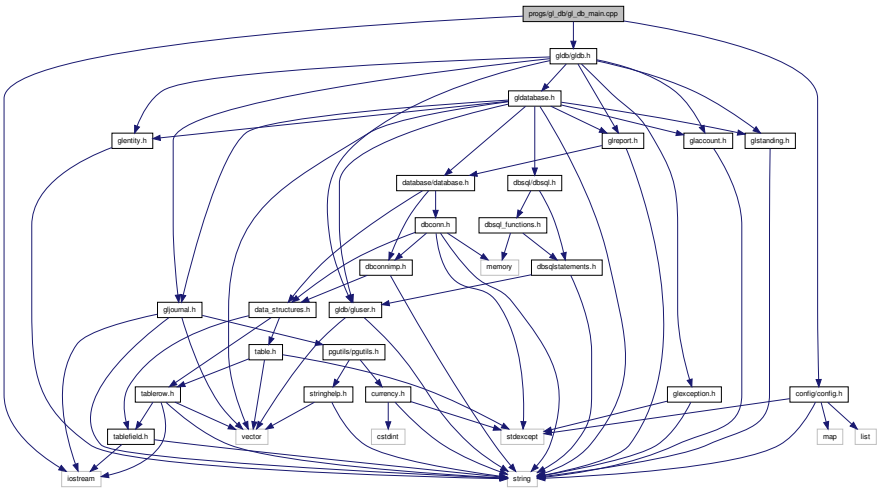
Copyright

10.52 progs/gl_db/gl_db_main.cpp File Reference

Main functionality for gl_db program.

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

Include dependency graph for gl_db_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 * progname = "gl_db"

  Static variable for program name.

10.52.1 Detailed Description

Main functionality for gl_db program.

Author

Paul Griffiths

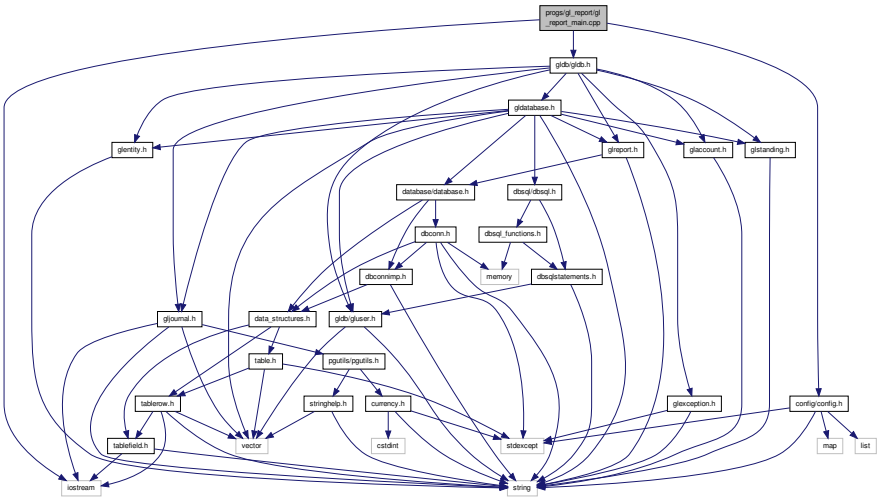
Copyright

10.53 progs/gl_report/gl_report_main.cpp File Reference

Main functionality for gl_report program.

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

Include dependency graph for gl_report_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 ∗ progname = "gl_report"

  Static variable for program name.

10.53.1 Detailed Description

Main functionality for gl_report program.

Author

Paul Griffiths

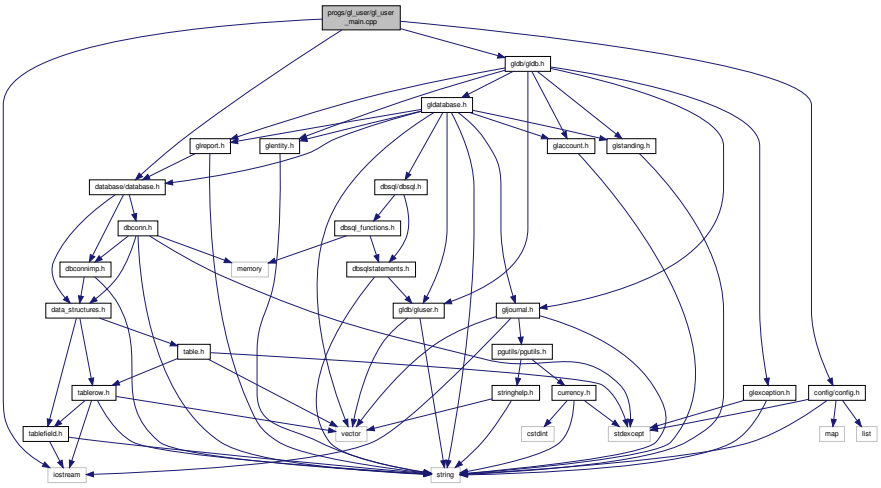
Copyright

10.54 progs/gl_user/gl_user_main.cpp File Reference

Main functionality for gl_user program.

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

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.
- GLUser get_user (Config &config, GLDatabase &gdb)

  Returns a user from either an ID or a name.
- static void show_user_details (const GLUser &user)

  Outputs details for a user.
- static void enable_user (GLUser &user, Config &config, GLDatabase &gdb)

  Enables or disables a user.
- static void set_user_password (GLUser &user, Config &config, GLDatabase &gdb)

  Sets a user's password.
- static void check_user_password (GLUser &user, Config &config)

  Checks a user's password.
- static void print_usage_message ()

  Prints a program usage message.

- static void print_version_message ()

  Prints a program version message.
- static void print_help_message ()

  Prints a program help message.
- static std::string login (void)

  Gets a password from the terminal.
- int main (int argc, char ∗argv[])

  Main function.

Variables

- static const char ∗ progname = "gl_user"

  Static variable for program name.

10.54.1 Detailed Description

Main functionality for gl_user program.

Author

Paul Griffiths

Copyright

Index

~Config
genleg::Config, 39
~DBConnDummy
gldb::DBConnDummy, 57
~DBConnImp
gldb::DBConnImp, 59
~DBConnMySQL
gldb::DBConnMySQL, 62
~DBSQLStatements
genleg::DBSQLStatements, 67
~GLDBTransaction
genleg::GLDBTransaction, 83
~GLDatabase
genleg::GLDatabase, 76
~GLUser
genleg::GLUser, 93
~MySQLResult
gldb::MySQLResult, 96
~Table
gldb::Table, 100
~TableField
gldb::TableField, 109
~TableRow
gldb::TableRow, 118
_XOPEN_SOURCE
config_getopt.cpp, 126
gluser_pass.cpp, 179

account
genleg::GLJELine, 86
account_by_name
genleg::DBSQLStatements, 67
add_cmdline_option
genleg::Config, 40
amount
genleg::GLJELine, 86
append_field
gldb::TableRow, 118, 119
append_record
gldb::Table, 100

backend
genleg::GLDatabase, 76
begin
genleg::GLJournal, 87, 88
gldb::Table, 100
gldb::TableRow, 119
boolstring_to_bool
gldatabase.cpp, 162

check_db_parameters
Database program., 32
Reporting program., 34
User administration program., 36
check_help_and_version
Database program., 32
Reporting program., 34
User administration program., 37
check_password
genleg::GLUser, 93
check_user_password
User administration program., 37
Config
genleg::Config, 39
config_getopt.cpp
_XOPEN_SOURCE, 126
ConfigBadConfigFile
genleg::ConfigBadConfigFile, 42
ConfigBadOption
genleg::ConfigBadOption, 43
ConfigCouldNotOpenFile
genleg::ConfigCouldNotOpenFile, 45
ConfigException
genleg::ConfigException, 46
ConfigOptionNotSet
genleg::ConfigOptionNotSet, 47
content_lines
General purpose utilities., 26
create_entity
genleg::GLDatabase, 76
create_from_file
gldb::Table, 100
create_structure
genleg::GLDatabase, 76
create_table
genleg::DBSQLStatements, 67
create_user
genleg::GLDatabase, 76
create_view
genleg::DBSQLStatements, 68
Currency
pgutils::Currency, 48
currency_from_string
General purpose utilities., 26
CurrencyException
pgutils::CurrencyException, 50
current_trial_balance_report
genleg::GLDatabase, 77
currenttb

genleg::DBSQLStatements, 68
currenttb_by_entity
genleg::DBSQLStatements, 68

DBConn
gldb::DBConn, 52
DBConnCouldNotConnect
gldb::DBConnCouldNotConnect, 54
DBConnCouldNotQuery
gldb::DBConnCouldNotQuery, 55
DBConnDummy
gldb::DBConnDummy, 56, 57
DBConnException
gldb::DBConnException, 58
DBConnImp
gldb::DBConnImp, 59
DBConnMySQL
gldb::DBConnMySQL, 62
DBSQLStatements
genleg::DBSQLStatements, 67
Database interaction module, 18
get_connection, 19
get_database_type, 19
get_field_names, 19
get_row, 19
Database program., 32
check_db_parameters, 32
check_help_and_version, 32
login, 33
main, 33
set_configuration, 33
decorated_report_from_table
General Ledger database module., 23
decorated_row
General Ledger database module., 23
destroy_structure
genleg::GLDatabase, 77
drop_table
genleg::DBSQLStatements, 68
drop_view
genleg::DBSQLStatements, 68

enable_user
User administration program., 37
enabled
genleg::GLUser, 93
end
genleg::GLJournal, 88
gldb::Table, 101
gldb::TableRow, 119
entity_by_id
genleg::DBSQLStatements, 69
entity_by_name
genleg::DBSQLStatements, 69
expand
pgutils::Currency, 48

firstname
genleg::GLUser, 93

GLAccount
genleg::GLAccount, 73
GLDBException
genleg::GLDBException, 81
GLDBTransaction
genleg::GLDBTransaction, 82
GLDatabase
genleg::GLDatabase, 75
GLEntity
genleg::GLEntity, 84
GLJELine
genleg::GLJELine, 85
GLJournal
genleg::GLJournal, 87
GLReport
genleg::GLReport, 89
GLStandingData
genleg::GLStandingData, 90
GLUser
genleg::GLUser, 92
General Ledger database module., 22
decorated_report_from_table, 23
decorated_row, 23
grow_widths, 23
max_column_widths, 23
plain_report_from_table, 23
plain_row, 24
separator_row, 24
General purpose utilities., 25
content_lines, 26
currency_from_string, 26
join, 26
next_content_line, 26
operator<, 27
operator<=, 28
operator>, 28
operator>=, 28
operator+, 27
operator-, 27
operator==, 28
replace, 29
split, 29
split_lines, 30
trim, 30
trim_back, 30
trim_front, 30
generate_salt
gluser_pass.cpp, 179
genleg::Config, 39
∼Config, 39
add_cmdline_option, 40
Config, 39
is_set, 40
m_opts_set, 41
m_opts_supp, 41
populate_from_cmdline, 40
populate_from_file, 41
genleg::ConfigBadConfigFile, 41

ConfigBadConfigFile, 42
genleg::ConfigBadOption, 43
ConfigBadOption, 43
genleg::ConfigCouldNotOpenFile, 44
ConfigCouldNotOpenFile, 45
genleg::ConfigException, 45
ConfigException, 46
genleg::ConfigOptionNotSet, 46
ConfigOptionNotSet, 47
genleg::DBSQLDummy, 64
genleg::DBSQLMySQL, 64
genleg::DBSQLStatements, 65
~DBSQLStatements, 67
account_by_name, 67
create_table, 67
create_view, 68
currenttb, 68
currenttb_by_entity, 68
DBSQLStatements, 67
drop_table, 68
drop_view, 68
entity_by_id, 69
entity_by_name, 69
get_perms, 69
grant, 69
je_by_id, 70
jelines_by_id, 70
listusers, 70
post_je, 70
post_je_line, 71
revoke, 71
standing_data, 71
update_user, 71
user_by_id, 71
user_by_username, 72
genleg::GLAccount, 72
GLAccount, 73
m_description, 73
m_enabled, 73
genleg::GLDBException, 81
GLDBException, 81
genleg::GLDBTransaction, 82
~GLDBTransaction, 83
GLDBTransaction, 82
m_commit, 83
m_dbc, 83
genleg::GLDatabase, 73
~GLDatabase, 76
backend, 76
create_entity, 76
create_structure, 76
create_user, 76
current_trial_balance_report, 77
destroy_structure, 77
GLDatabase, 75
get_account_by_name, 77
get_entity_by_id, 77
get_entity_by_name, 78
get_je_by_id, 78
get_standing_data, 78
get_user_by_id, 78
get_user_by_username, 79
grant, 79
je_report, 79
list_users_report, 79
load_sample_data, 79
m_dbc, 80
m_sql, 81
m_tables, 81
m_views, 81
post_journal, 80
report, 80
revoke, 80
standing_data_report, 80
update_user, 80
genleg::GLEntity, 83
GLEntity, 84
m_aggregate, 84
m_enabled, 84
m_name, 84
m_parent, 84
m_shortname, 84
genleg::GLJELine, 85
account, 86
amount, 86
GLJELine, 85
m_acct, 86
m_amount, 86
genleg::GLJournal, 86
begin, 87, 88
end, 88
GLJournal, 87
m_entity, 88
m_id, 88
m_lines, 88
m_memo, 88
m_period, 88
m_source, 88
m_user, 89
m_year, 89
genleg::GLReport, 89
GLReport, 89
m_headers, 89
m_report_text, 89
m_title, 90
genleg::GLStandingData, 90
GLStandingData, 90
m_num_periods, 90
m_period, 90
m_year, 91
genleg::GLUser, 91
~GLUser, 93
check_password, 93
enabled, 93
firstname, 93
GLUser, 92

id, 93
lastname, 93
m_enabled, 95
m_firstname, 95
m_id, 95
m_lastname, 95
m_pass_hash, 95
m_pass_salt, 95
m_perms, 95
m_username, 95
pass_hash, 93
pass_salt, 94
permissions, 94
set_enabled, 94
set_firstname, 94
set_lastname, 94
set_password, 94
set_username, 95
username, 95
get_account_by_name
genleg::GLDatabase, 77
get_connection
Database interaction module, 19
get_database_type
Database interaction module, 19
get_entity_by_id
genleg::GLDatabase, 77
get_entity_by_name
genleg::GLDatabase, 78
get_field
gldb::Table, 101
get_field_names
Database interaction module, 19
get_headers
gldb::Table, 101
get_je_by_id
genleg::GLDatabase, 78
get_perms
genleg::DBSQLStatements, 69
get_row
Database interaction module, 19
get_standing_data
genleg::GLDatabase, 78
get_user
User administration program., 37
get_user_by_id
genleg::GLDatabase, 78
get_user_by_username
genleg::GLDatabase, 79
gldatabase.cpp
boolstring_to_bool, 162
gldb::DBConn, 50
DBConn, 52
last_auto_increment, 52
m_imp, 53
operator=, 52
query, 52
select, 52
gldb::DBConnCouldNotConnect, 53
DBConnCouldNotConnect, 54
gldb::DBConnCouldNotQuery, 54
DBConnCouldNotQuery, 55
gldb::DBConnDummy, 55
∼DBConnDummy, 57
DBConnDummy, 56, 57
operator=, 57
query, 57
select, 57
gldb::DBConnException, 58
DBConnException, 58
gldb::DBConnImp, 58
∼DBConnImp, 59
DBConnImp, 59
last_auto_increment, 60
query, 60
select, 60
gldb::DBConnMySQL, 60
∼DBConnMySQL, 62
DBConnMySQL, 62
last_auto_increment, 62
m_conn, 63
mtx, 63
operator=, 62, 63
query, 63
select, 63
gldb::MySQLResult, 96
∼MySQLResult, 96
m_num_fields, 97
m_result, 97
MySQLResult, 96, 97
num_fields, 97
operator=, 97
result, 97
gldb::Table, 98
∼Table, 100
append_record, 100
begin, 100
create_from_file, 100
end, 101
get_field, 101
get_headers, 101
insert_query, 102
m_headers, 103
m_quoted, 103
m_records, 103
num_fields, 102
num_records, 102
operator=, 102
set_quoted, 103
Table, 99, 100
gldb::TableBadInputFile, 104
TableBadInputFile, 104
gldb::TableCouldNotOpenInputFile, 105
TableCouldNotOpenInputFile, 106
gldb::TableException, 106
TableException, 107

gldb::TableField, 107
~TableField, 109
length, 109
m_data, 112
operator std::string, 109
operator<<, 112
operator+=, 109, 110
operator=, 110, 111
TableField, 108, 109
gldb::TableMismatchedRecordLength, 112
TableMismatchedRecordLength, 113
gldb::TableNoSuchField, 113
TableNoSuchField, 114
gldb::TableNoSuchRecord, 115
TableNoSuchRecord, 115
gldb::TableRow, 116
~TableRow, 118
append_field, 118, 119
begin, 119
end, 119
m_fields, 121
operator=, 120
print, 120
record_string, 121
size, 121
TableRow, 117, 118
gluser_pass.cpp
_XOPEN_SOURCE, 179
generate_salt, 179
grant
genleg::DBSQLStatements, 69
genleg::GLDatabase, 79
grow_widths
General Ledger database module., 23

id
genleg::GLUser, 93
insert_query
gldb::Table, 102
is_set
genleg::Config, 40

je_by_id
genleg::DBSQLStatements, 70
je_report
genleg::GLDatabase, 79
jelines_by_id
genleg::DBSQLStatements, 70
join
General purpose utilities., 26

last_auto_increment
gldb::DBConn, 52
gldb::DBConnImp, 60
gldb::DBConnMySQL, 62
lastname
genleg::GLUser, 93
length
gldb::TableField, 109
lib/config/config.cpp, 123
lib/config/config.h, 124
lib/config/config_getopt.cpp, 125
lib/database/data_structures.h, 126
lib/database/database.h, 127
lib/database/dbconn.cpp, 129
lib/database/dbconn.h, 130
lib/database/dbconnimp.h, 131
lib/database/table.cpp, 133
lib/database/table.h, 134
lib/database/tablefield.cpp, 136
lib/database/tablefield.h, 136
lib/database/tablerow.cpp, 138
lib/database/tablerow.h, 139
lib/database_imp/database_imp.h, 140
lib/database_imp/dummy/dbconn_dummy_imp.cpp, 142
lib/database_imp/dummy/dbconn_dummy_imp.h, 143
lib/database_imp/mysql/dbconn_mysql_functions.cpp, 145
lib/database_imp/mysql/dbconn_mysql_imp.cpp, 146
lib/database_imp/mysql/dbconn_mysql_imp.h, 148
lib/database_imp/mysql/dbconn_mysql_result.cpp, 149
lib/database_imp/mysql/dbconn_mysql_result.h, 150
lib/dbsql/dbsql.h, 151
lib/dbsql/dbsql_dummy.h, 152
lib/dbsql/dbsql_functions.h, 154
lib/dbsql/dbsql_implementations.h, 155
lib/dbsql/dbsql_mysql.h, 157
lib/dbsql/dbsqlstatements.cpp, 158
lib/dbsql/dbsqlstatements.h, 159
lib/gldb/glaccount.cpp, 160
lib/gldb/glaccount.h, 160
lib/gldb/gldatabase.cpp, 162
lib/gldb/gldatabase.h, 163
lib/gldb/gldb.h, 164
lib/gldb/glentity.cpp, 165
lib/gldb/glentity.h, 166
lib/gldb/glexception.h, 167
lib/gldb/gljournal.cpp, 169
lib/gldb/gljournal.h, 169
lib/gldb/glreport.cpp, 171
lib/gldb/glreport.h, 172
lib/gldb/glstanding.cpp, 174
lib/gldb/glstanding.h, 175
lib/gldb/gluser.cpp, 176
lib/gldb/gluser.h, 177
lib/gldb/gluser_pass.cpp, 178
lib/pgutils/currency.cpp, 179
lib/pgutils/currency.h, 180
lib/pgutils/pgutils.h, 182
lib/pgutils/stringhelp.cpp, 183
lib/pgutils/stringhelp.h, 183
list_users_report
genleg::GLDatabase, 79
listusers
genleg::DBSQLStatements, 70
load_sample_data
genleg::GLDatabase, 79

login
Database program., 33
Reporting program., 35
User administration program., 37

m_acct
genleg::GLJELine, 86
m_aggregate
genleg::GLEntity, 84
m_amount
genleg::GLJELine, 86
m_commit
genleg::GLDBTransaction, 83
m_conn
gldb::DBConnMySQL, 63
m_data
gldb::TableField, 112
m_dbc
genleg::GLDatabase, 80
genleg::GLDBTransaction, 83
m_description
genleg::GLAccount, 73
m_enabled
genleg::GLAccount, 73
genleg::GLEntity, 84
genleg::GLUser, 95
m_entity
genleg::GLJournal, 88
m_fields
gldb::TableRow, 121
m_firstname
genleg::GLUser, 95
m_frac
pgutils::Currency, 50
m_headers
genleg::GLReport, 89
gldb::Table, 103
m_id
genleg::GLJournal, 88
genleg::GLUser, 95
m_imp
gldb::DBConn, 53
m_int
pgutils::Currency, 50
m_lastname
genleg::GLUser, 95
m_lines
genleg::GLJournal, 88
m_memo
genleg::GLJournal, 88
m_name
genleg::GLEntity, 84
m_num_fields
gldb::MySQLResult, 97
m_num_periods
genleg::GLStandingData, 90
m_opts_set
genleg::Config, 41
m_opts_supp
genleg::Config, 41
m_parent
genleg::GLEntity, 84
m_pass_hash
genleg::GLUser, 95
m_pass_salt
genleg::GLUser, 95
m_period
genleg::GLJournal, 88
genleg::GLStandingData, 90
m_perms
genleg::GLUser, 95
m_quoted
gldb::Table, 103
m_records
gldb::Table, 103
m_report_text
genleg::GLReport, 89
m_result
gldb::MySQLResult, 97
m_shortname
genleg::GLEntity, 84
m_source
genleg::GLJournal, 88
m_sql
genleg::GLDatabase, 81
m_tables
genleg::GLDatabase, 81
m_title
genleg::GLReport, 90
m_user
genleg::GLJournal, 89
m_username
genleg::GLUser, 95
m_views
genleg::GLDatabase, 81
m_year
genleg::GLJournal, 89
genleg::GLStandingData, 91
main
Database program., 33
Reporting program., 35
User administration program., 38
max_column_widths
General Ledger database module., 23
mtx
gldb::DBConnMySQL, 63
MySQLResult
gldb::MySQLResult, 96, 97

next_content_line
General purpose utilities., 26
num_fields
gldb::MySQLResult, 97
gldb::Table, 102
num_records
gldb::Table, 102

operator std::string

gldb::TableField, 109
operator<
General purpose utilities., 27
pgutils::Currency, 49
operator<<
gldb::TableField, 112
operator<=
General purpose utilities., 28
operator>
General purpose utilities., 28
operator>=
General purpose utilities., 28
operator+
General purpose utilities., 27
pgutils::Currency, 49
operator+=
gldb::TableField, 109, 110
pgutils::Currency, 48
operator-
General purpose utilities., 27
pgutils::Currency, 48
operator-=
pgutils::Currency, 48
operator=
gldb::DBConn, 52
gldb::DBConnDummy, 57
gldb::DBConnMySQL, 62, 63
gldb::MySQLResult, 97
gldb::Table, 102
gldb::TableField, 110, 111
gldb::TableRow, 120
operator==
General purpose utilities., 28
pgutils::Currency, 49

pass_hash
genleg::GLUser, 93
pass_salt
genleg::GLUser, 94
permissions
genleg::GLUser, 94
pgutils::Currency, 47
Currency, 48
expand, 48
m_frac, 50
m_int, 50
operator<, 49
operator+, 49
operator+=, 48
operator-, 48
operator-=, 48
operator==, 49
pgutils::CurrencyException, 50
CurrencyException, 50
plain_report_from_table
General Ledger database module., 23
plain_row
General Ledger database module., 24
populate_from_cmdline
genleg::Config, 40
populate_from_file
genleg::Config, 41
post_je
genleg::DBSQLStatements, 70
post_je_line
genleg::DBSQLStatements, 71
post_journal
genleg::GLDatabase, 80
print
gldb::TableRow, 120
Program configuration module, 17
progs/gl_db/gl_db_main.cpp, 185
progs/gl_report/gl_report_main.cpp, 186
progs/gl_user/gl_user_main.cpp, 188

query
gldb::DBConn, 52
gldb::DBConnDummy, 57
gldb::DBConnImp, 60
gldb::DBConnMySQL, 63

record_string
gldb::TableRow, 121
replace
General purpose utilities., 29
report
genleg::GLDatabase, 80
Reporting program., 34
check_db_parameters, 34
check_help_and_version, 34
login, 35
main, 35
set_configuration, 35
result
gldb::MySQLResult, 97
revoke
genleg::DBSQLStatements, 71
genleg::GLDatabase, 80

SQL statements module, 21
select
gldb::DBConn, 52
gldb::DBConnDummy, 57
gldb::DBConnImp, 60
gldb::DBConnMySQL, 63
separator_row
General Ledger database module., 24
set_configuration
Database program., 33
Reporting program., 35
User administration program., 38
set_enabled
genleg::GLUser, 94
set_firstname
genleg::GLUser, 94
set_lastname
genleg::GLUser, 94
set_password

genleg::GLUser, 94
set_quoted
gldb::Table, 103
set_user_password
User administration program., 38
set_username
genleg::GLUser, 95
show_user_details
User administration program., 38
size
gldb::TableRow, 121
split
General purpose utilities., 29
split_lines
General purpose utilities., 30
standing_data
genleg::DBSQLStatements, 71
standing_data_report
genleg::GLDatabase, 80

Table
gldb::Table, 99, 100
TableBadInputFile
gldb::TableBadInputFile, 104
TableCouldNotOpenInputFile
gldb::TableCouldNotOpenInputFile, 106
TableException
gldb::TableException, 107
TableField
gldb::TableField, 108, 109
TableMismatchedRecordLength
gldb::TableMismatchedRecordLength, 113
TableNoSuchField
gldb::TableNoSuchField, 114
TableNoSuchRecord
gldb::TableNoSuchRecord, 115
TableRow
gldb::TableRow, 117, 118
trim
General purpose utilities., 30
trim_back
General purpose utilities., 30
trim_front
General purpose utilities., 30

update_user
genleg::DBSQLStatements, 71
genleg::GLDatabase, 80
User administration program., 36
check_db_parameters, 36
check_help_and_version, 37
check_user_password, 37
enable_user, 37
get_user, 37
login, 37
main, 38
set_configuration, 38
set_user_password, 38
show_user_details, 38
user_by_id
genleg::DBSQLStatements, 71
user_by_username
genleg::DBSQLStatements, 72
username
genleg::GLUser, 95