

Apos - Database Manager

Generated on Fri Nov 17 2023 18:52:30 for Apos - Database Manager by Doxygen 1.9.8

Fri Nov 17 2023 18:52:30

1 Apos - Database Manager	1
1.1 Features	1
1.2 Prerequisites	1
1.3 Installation	1
1.4 Usage	2
1.5 File Structure	2
1.6 Documentation	2
1.7 Contributing	2
1.8 License	2
1.9 Acknowledgments	2
2 Namespace Index	3
2.1 Namespace List	3
3 Hierarchical Index	5
3.1 Class Hierarchy	5
4 Class Index	7
4.1 Class List	7
5 File Index	9
5.1 File List	9
6 Namespace Documentation	11
6.1 Ui Namespace Reference	11
7 Class Documentation	13
7.1 DatabaseHandler Class Reference	13
7.1.1 Detailed Description	13
7.1.2 Constructor & Destructor Documentation	13
7.1.2.1 DatabaseHandler()	13
7.1.3 Member Function Documentation	13
7.1.3.1 closeDatabase()	13
7.1.3.2 executeCommand()	14
7.1.3.3 getActiveDatabase()	14
7.1.3.4 getSqlError()	14
7.1.3.5 initDatabase()	15
7.1.3.6 setDatabasePath()	15
7.1.3.7 setLastSqlError()	15
7.1.4 Member Data Documentation	15
7.1.4.1 activeDatabase	15
7.1.4.2 databasePath	15
7.1.4.3 lastSqlError	15
7.2 DevWindow Class Reference	16

7.2.1 Detailed Description	17
7.2.2 Constructor & Destructor Documentation	17
7.2.2.1 DevWindow()	17
7.2.2.2 ~DevWindow()	17
7.2.3 Member Function Documentation	18
7.2.3.1 assignInputs()	18
7.2.3.2 checkCheckbox()	18
7.2.3.3 clearCommandBox()	18
7.2.3.4 clearInputs()	19
7.2.3.5 closeDatabase()	20
7.2.3.6 enableButtons()	20
7.2.3.7 initDatabase()	21
7.2.3.8 logEvent() [1/3]	21
7.2.3.9 logEvent() [2/3]	22
7.2.3.10 logEvent() [3/3]	22
7.2.3.11 on_btnAdd_clicked	22
7.2.3.12 on_btnCloseDB_clicked	23
7.2.3.13 on_btnExecute_clicked	23
7.2.3.14 on_btnInitDB_clicked	24
7.2.3.15 on_btnSelectTable_clicked	24
7.2.3.16 on_btnUpdate_clicked	25
7.2.3.17 on_clearCommandAfterExecute_stateChanged	25
7.2.3.18 on_clearInputsAfterInsert_stateChanged	26
7.2.3.19 on_inReturnToLauncher_clicked	26
7.2.3.20 on_inSettings_clicked	26
7.2.3.21 openSettings	27
7.2.3.22 retranslateUi()	27
7.2.3.23 returnToLauncher	27
7.2.3.24 setModelViews() [1/2]	27
7.2.3.25 setModelViews() [2/2]	28
7.2.4 Member Data Documentation	28
7.2.4.1 clearCommand	28
7.2.4.2 clearInput	28
7.2.4.3 input1	28
7.2.4.4 input2	28
7.2.4.5 input3	28
7.2.4.6 input4	29
7.2.4.7 input5	29
7.2.4.8 objectHandler	29
7.2.4.9 ui	29
7.3 LauncherWindow Class Reference	29
7.3.1 Detailed Description	30

7.3.2 Constructor & Destructor Documentation	30
7.3.2.1 LauncherWindow()	30
7.3.2.2 ~LauncherWindow()	31
7.3.3 Member Function Documentation	31
7.3.3.1 on_inShowDev_clicked	31
7.3.3.2 on_pushButton_clicked	31
7.3.3.3 openDevWindow	31
7.3.3.4 openSettings	32
7.3.3.5 retranslateUi()	32
7.3.4 Member Data Documentation	32
7.3.4.1 objectHandler	32
7.3.4.2 ui	32
7.4 ObjectHandler Class Reference	32
7.4.1 Detailed Description	33
7.4.2 Constructor & Destructor Documentation	33
7.4.2.1 ObjectHandler()	33
7.4.3 Member Function Documentation	33
7.4.3.1 getActiveDatabase()	33
7.4.3.2 getActiveTableName()	34
7.4.3.3 getDbHandler()	34
7.4.3.4 getTableHandler()	35
7.4.3.5 getTableSqlError()	35
7.4.3.6 initDatabaseObject()	36
7.4.3.7 initTableObject() [1/2]	36
7.4.3.8 initTableObject() [2/2]	37
7.4.3.9 setActiveTableName()	37
7.4.4 Member Data Documentation	37
7.4.4.1 application	37
7.4.4.2 dbHandler	38
7.4.4.3 tableHandler	38
7.5 SettingsWindow Class Reference	38
7.5.1 Detailed Description	39
7.5.2 Constructor & Destructor Documentation	39
7.5.2.1 SettingsWindow()	39
7.5.2.2 ~SettingsWindow()	39
7.5.3 Member Function Documentation	39
7.5.3.1 appliedSettings	39
7.5.3.2 installTranslator()	40
7.5.3.3 on_inApply_clicked	40
7.5.3.4 on_inClose_clicked	40
7.5.3.5 on_inLanguage_currentIndexChanged	41
7.5.3.6 retranslateUi()	41

7.5.4 Member Data Documentation	41
7.5.4.1 language	41
7.5.4.2 languageChanged	41
7.5.4.3 languageIndex	41
7.5.4.4 objectHandler	41
7.5.4.5 tempLanguageIndex	41
7.5.4.6 translator	41
7.5.4.7 ui	41
7.6 StartupHandler Class Reference	41
7.6.1 Detailed Description	42
7.6.2 Constructor & Destructor Documentation	42
7.6.2.1 StartupHandler()	42
7.6.2.2 ~StartupHandler()	42
7.6.3 Member Function Documentation	43
7.6.3.1 initObjectHandler()	43
7.6.3.2 initTranslator()	43
7.6.3.3 startUp()	43
7.6.4 Member Data Documentation	44
7.6.4.1 objectHandler	44
7.6.4.2 ptrApplication	44
7.7 TableHandler Class Reference	44
7.7.1 Detailed Description	45
7.7.2 Constructor & Destructor Documentation	45
7.7.2.1 TableHandler() [1/2]	45
7.7.2.2 TableHandler() [2/2]	45
7.7.3 Member Function Documentation	45
7.7.3.1 generateTableModel() [1/2]	45
7.7.3.2 generateTableModel() [2/2]	46
7.7.3.3 getActiveTableName()	46
7.7.3.4 getLastTableError()	46
7.7.3.5 getTableModel()	46
7.7.3.6 getTableSQLException()	47
7.7.3.7 getTblActiveDatabase()	47
7.7.3.8 insertIntoTable()	47
7.7.3.9 setActiveDatabase()	47
7.7.3.10 setActiveTableName()	47
7.7.3.11 setTableName()	47
7.7.4 Member Data Documentation	48
7.7.4.1 activeTableName	48
7.7.4.2 lastTableError	48
7.7.4.3 tableModel	48
7.7.4.4 tblActiveDatabase	48

7.8 TranslatableWindow Class Reference	48
7.8.1 Detailed Description	48
7.8.2 Constructor & Destructor Documentation	48
7.8.2.1 TranslatableWindow()	48
7.8.3 Member Function Documentation	49
7.8.3.1 retranslateUi()	49
7.9 WindowHandler Class Reference	49
7.9.1 Detailed Description	50
7.9.2 Constructor & Destructor Documentation	50
7.9.2.1 WindowHandler()	50
7.9.3 Member Function Documentation	50
7.9.3.1 applySettings	50
7.9.3.2 changeLanguages()	51
7.9.3.3 showDevWindow	51
7.9.3.4 showLaunchWindow	52
7.9.3.5 showSettingsWindow	52
7.9.4 Member Data Documentation	53
7.9.4.1 devWindow	53
7.9.4.2 launcherWindow	53
7.9.4.3 objectHandler	53
7.9.4.4 settingsWindow	53
8 File Documentation	55
8.1 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/backendClasses/objecthandler.cpp	
File Reference	55
8.2 objecthandler.cpp	55
8.3 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/backendClasses/objecthandler.hpp	
File Reference	56
8.4 objecthandler.hpp	57
8.5 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/backendClasses/startuphandler.cpp	
File Reference	58
8.6 startuphandler.cpp	58
8.7 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/backendClasses/startuphandler.hpp	
File Reference	59
8.8 startuphandler.hpp	60
8.9 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/databaseClasses/databasehandler.cpp	
File Reference	61
8.10 databasehandler.cpp	61
8.11 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/databaseClasses/databasehandler.hpp	
File Reference	62
8.12 databasehandler.hpp	63
8.13 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/databaseClasses/tablehandler.cpp	
File Reference	64
8.14 tablehandler.cpp	64

8.15 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/databaseClasses/tablehandler.hpp File Reference	65
8.16 tablehandler.hpp	66
8.17 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/devwindow.cpp File Reference	67
8.18 devwindow.cpp	67
8.19 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/devwindow.hpp File Reference	70
8.20 devwindow.hpp	71
8.21 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/launcherwindow.cpp File Reference	72
8.22 launcherwindow.cpp	72
8.23 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/launcherwindow.hpp File Reference	73
8.24 launcherwindow.hpp	74
8.25 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/settingswindow.cpp File Reference	75
8.26 settingswindow.cpp	75
8.27 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/settingswindow.hpp File Reference	76
8.28 settingswindow.hpp	77
8.29 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/translatablewindow.cpp File Reference	78
8.30 translatablewindow.cpp	78
8.31 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/translatablewindow.hpp File Reference	79
8.32 translatablewindow.hpp	79
8.33 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/windowhandler.cpp File Reference	79
8.34 windowhandler.cpp	80
8.35 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/windowhandler.hpp File Reference	81
8.36 windowhandler.hpp	82
8.37 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/README.md File Reference .	82
8.38 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/source/main.cpp File Reference	82
8.38.1 Function Documentation	83
8.38.1.1 main()	83
8.39 main.cpp	83

Index

85

Chapter 1

Apos - Database Manager

Apos - Database Manager is a Windows application designed to manage SQLite3 databases. It allows users to open, read, and edit tables within a SQLite3 database, as well as execute custom SQLite3 commands.

1.1 Features

- **Database Operations:**
 - Open and manage SQLite3 databases.
 - Execute custom SQLite3 commands.
 - Read and edit tables within the database.
- **User Interface:**
 - Graphical User Interface (GUI) created using the Qt framework.
 - Intuitive design for easy navigation.

1.2 Prerequisites

- Windows Operating System

1.3 Installation

The software is installed using a custom installer available in the [Releases](#) section.

1. Download the installer from the Releases page.
2. Run the installer and follow the on-screen instructions.

1.4 Usage

1. Opening a Database:

- Launch the application.
- Click on the "Open Database" button.
- Select the desired SQLite3 database file.

2. Executing Commands:

- Use the "Command Box" to enter custom SQLite3 commands.
- Click the "Execute" button to execute the command.

3. Managing Tables:

- Use the GUI to read and edit tables within the opened database.
- Buttons like "Add," "Update," and "Select Table" facilitate table operations.

1.5 File Structure

- `/source`: Contains the source code for Apos - Database Manager.
- `/basicClasses`: Basic classes used by the application.
- `/databaseClasses`: Database-related classes.
- `/tableClasses`: Classes related to table operations.
- `/resources`: Includes any additional resources used by the application.

1.6 Documentation

The full documentation for the application, including Doxygen-generated documentation, can be found [\[here\]](#)(link-to-documentation). - TBD

1.7 Contributing

If you would like to contribute to the project, please follow the guidelines outlined in CONTRIBUTING.md. - TBD

1.8 License

This project is licensed under the LGPL-3.0 License.

1.9 Acknowledgments

- Special thanks to the Qt framework community.
- [Any other acknowledgments or credits]

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Ui	11
--------------------------	----

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

DatabaseHandler	13
ObjectHandler	32
QMainWindow	
DevWindow	16
LauncherWindow	29
QObject	
WindowHandler	49
QWidget	
SettingsWindow	38
StartupHandler	41
TableHandler	44
TranslatableWindow	48
DevWindow	16
LauncherWindow	29
SettingsWindow	38

Chapter 4

Class Index

4.1 Class List

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

DatabaseHandler	13
DevWindow	16
LauncherWindow	29
ObjectHandler	32
SettingsWindow	38
StartupHandler	41
TableHandler	44
TranslatableWindow	48
WindowHandler	49

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/backendClasses/objecthandler.cpp	55
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/backendClasses/objecthandler.hpp	56
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/backendClasses/startuphandler.cpp	58
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/backendClasses/startuphandler.hpp	59
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/databaseClasses/databasehandler.cpp	61
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/databaseClasses/databasehandler.hpp	62
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/databaseClasses/tablehandler.cpp	64
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/databaseClasses/tablehandler.hpp	65
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/devwindow.cpp	67
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/devwindow.hpp	70
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/launcherwindow.cpp	72
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/launcherwindow.hpp	73
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/settingswindow.cpp	75
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/settingswindow.hpp	76
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/translatablewindow.cpp	78
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/translatablewindow.hpp	79
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/windowhandler.cpp	79
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/windowhandler.hpp	81
C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/source/main.cpp	82

Chapter 6

Namespace Documentation

6.1 Ui Namespace Reference

Chapter 7

Class Documentation

7.1 DatabaseHandler Class Reference

```
#include <databasehandler.hpp>
```

Public Member Functions

- [DatabaseHandler](#) ()
- bool [initDatabase](#) ()
- void [closeDatabase](#) ()
- bool [executeCommand](#) (QString command)
- const QSqlDatabase & [getActiveDatabase](#) () const
- const QSqlError & [getSqlError](#) () const
- void [setDatabasePath](#) (const QString &newDatabasePath)
- void [setLastSqlError](#) (const QSqlError &newLastSqlError)

Private Attributes

- QSqlError [lastSqlError](#)
- QSqlDatabase [activeDatabase](#)
- QString [databasePath](#) = R"(C:\Users\Clean\Documents\Projekte\Apos-DatabaseManager\Project\resources\defaultDatabase\userDatabase.db)"

7.1.1 Detailed Description

Definition at line 7 of file [databasehandler.hpp](#).

7.1.2 Constructor & Destructor Documentation

7.1.2.1 DatabaseHandler()

```
DatabaseHandler::DatabaseHandler ( )
```

Definition at line 3 of file [databasehandler.cpp](#).

7.1.3 Member Function Documentation

7.1.3.1 closeDatabase()

```
void DatabaseHandler::closeDatabase ( )
```

Definition at line 25 of file [databasehandler.cpp](#).

Here is the caller graph for this function:



7.1.3.2 executeCommand()

```
bool DatabaseHandler::executeCommand (
    QString command )
```

Definition at line 31 of file [databasehandler.cpp](#).

Here is the caller graph for this function:

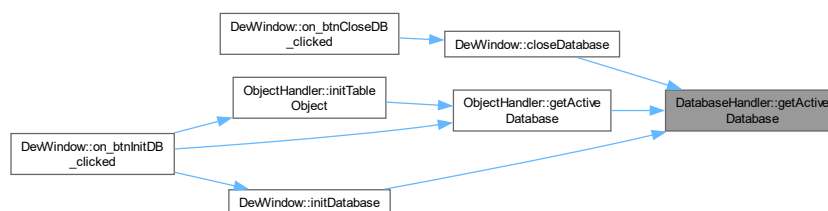


7.1.3.3 getActiveDatabase()

```
const QSqlDatabase & DatabaseHandler::getActiveDatabase ( ) const
```

Definition at line 43 of file [databasehandler.cpp](#).

Here is the caller graph for this function:

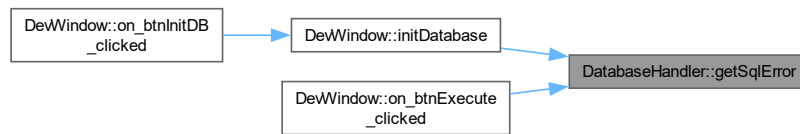


7.1.3.4 getSqlError()

```
const QSqlError & DatabaseHandler::getSqlError ( ) const
```

Definition at line 48 of file [databasehandler.cpp](#).

Here is the caller graph for this function:



7.1.3.5 initDatabase()

```
bool DatabaseHandler::initDatabase ( )
```

Definition at line 8 of file [databasehandler.cpp](#).

Here is the caller graph for this function:



7.1.3.6 setDatabasePath()

```
void DatabaseHandler::setDatabasePath (
    const QString & newDatabasePath )
```

Definition at line 53 of file [databasehandler.cpp](#).

7.1.3.7 setLastSqlError()

```
void DatabaseHandler::setLastSqlError (
    const QSqlError & newLastSqlError )
```

Definition at line 58 of file [databasehandler.cpp](#).

7.1.4 Member Data Documentation

7.1.4.1 activeDatabase

```
QSqlDatabase DatabaseHandler::activeDatabase [private]
```

Definition at line 39 of file [databasehandler.hpp](#).

7.1.4.2 databasePath

```
QString DatabaseHandler::databasePath = R"(C:\Users\Clean\Documents\Projekte\Apos-DatabaseManager\Project\resources\defaultDatabase\userDatabase.db)" [private]
```

Definition at line 41 of file [databasehandler.hpp](#).

7.1.4.3 lastSqlError

```
QSqlError DatabaseHandler::lastSqlError [private]
```

Definition at line 37 of file [databasehandler.hpp](#).

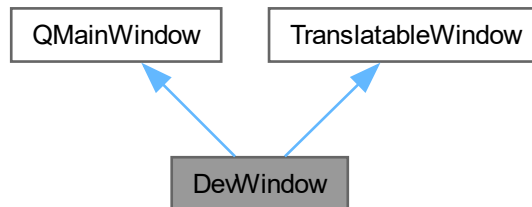
The documentation for this class was generated from the following files:

- C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/databaseClasses/[databasehandler.hpp](#)
- C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/databaseClasses/[databasehandler.cpp](#)

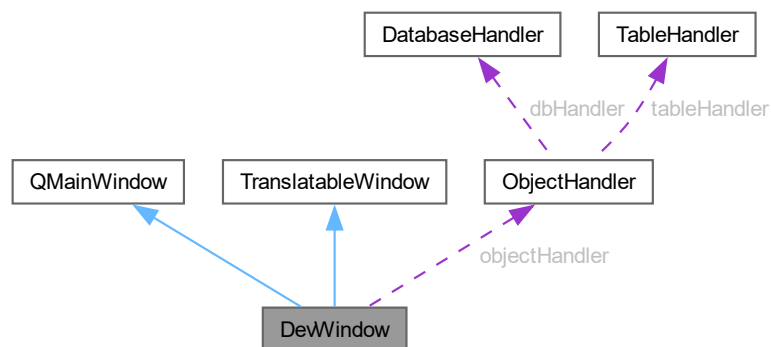
7.2 DevWindow Class Reference

```
#include <devwindow.hpp>
```

Inheritance diagram for DevWindow:



Collaboration diagram for DevWindow:



Signals

- void [returnToLauncher](#) ()
- void [openSettings](#) ()

Public Member Functions

- [DevWindow](#) (QWidget *parent=nullptr, [ObjectHandler](#) *objectHandler=nullptr)
- [~DevWindow](#) ()
- void [logEvent](#) (QString type, QString message)
- void [logEvent](#) (QString message, QSqlError error)
- void [logEvent](#) (QString message)
- void [retranslateUi](#) ()

Public Member Functions inherited from [TranslatableWindow](#)

- [TranslatableWindow](#) ()

Private Slots

- void [on_btnInitDB_clicked](#) ()
- void [on_btnCloseDB_clicked](#) ()
- void [on_btnExecute_clicked](#) ()
- void [on_btnSelectTable_clicked](#) ()
- void [on_btnAdd_clicked](#) ()
- void [on_btnUpdate_clicked](#) ()
- void [on_clearCommandAfterExecute_stateChanged](#) (int arg1)
- void [on_clearInputsAfterInsert_stateChanged](#) (int arg1)
- void [on_inReturnToLauncher_clicked](#) ()
- void [on_inSettings_clicked](#) ()

Private Member Functions

- void [enableButtons](#) (bool databaseLoaded)
- void [setModelViews](#) (QSqlTableModel &m)
- void [setModelViews](#) ()
- void [assignInputs](#) ()
- bool [checkCheckbox](#) (int argCB)
- void [initDatabase](#) (ObjectHandler *oH)
- void [closeDatabase](#) (DatabaseHandler *db)
- void [clearInputs](#) (bool clearBool)
- void [clearCommandBox](#) (bool clearBool)

Private Attributes

- Ui::DevWindow * [ui](#)
- ObjectHandler * [objectHandler](#)
- QString [input1](#)
- QString [input2](#)
- QString [input3](#)
- QString [input4](#)
- QString [input5](#)
- bool [clearCommand](#) = false
- bool [clearInput](#) = false

7.2.1 Detailed Description

Definition at line 14 of file [devwindow.hpp](#).

7.2.2 Constructor & Destructor Documentation**7.2.2.1 DevWindow()**

```
DevWindow::DevWindow (
    QWidget * parent = nullptr,
    ObjectHandler * objectHandler = nullptr )
```

Definition at line 6 of file [devwindow.cpp](#).

7.2.2.2 ~DevWindow()

```
DevWindow::~DevWindow ( )
```

Definition at line 14 of file [devwindow.cpp](#).

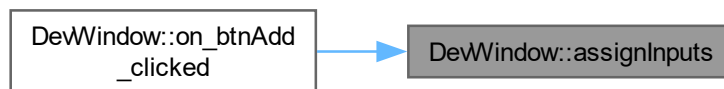
7.2.3 Member Function Documentation

7.2.3.1 assignInputs()

```
void DevWindow::assignInputs ( ) [private]
```

Definition at line 62 of file [devwindow.cpp](#).

Here is the caller graph for this function:

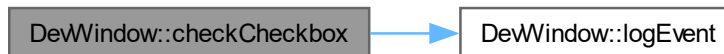


7.2.3.2 checkCheckbox()

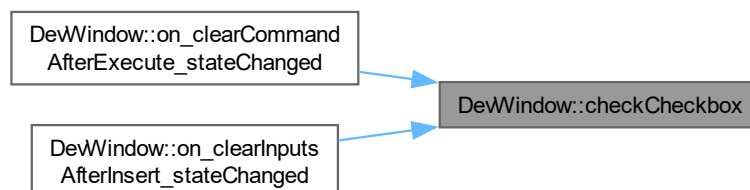
```
bool DevWindow::checkCheckbox (
    int argCB ) [private]
```

Definition at line 89 of file [devwindow.cpp](#).

Here is the call graph for this function:



Here is the caller graph for this function:

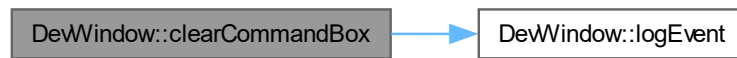


7.2.3.3 clearCommandBox()

```
void DevWindow::clearCommandBox (
    bool clearBool ) [private]
```

Definition at line 115 of file [devwindow.cpp](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.2.3.4 clearInputs()

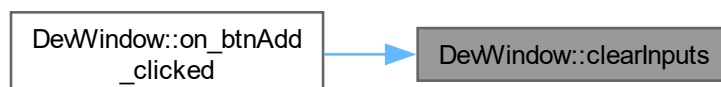
```
void DevWindow::clearInputs (
    bool clearBool ) [private]
```

Definition at line 103 of file [devwindow.cpp](#).

Here is the call graph for this function:



Here is the caller graph for this function:

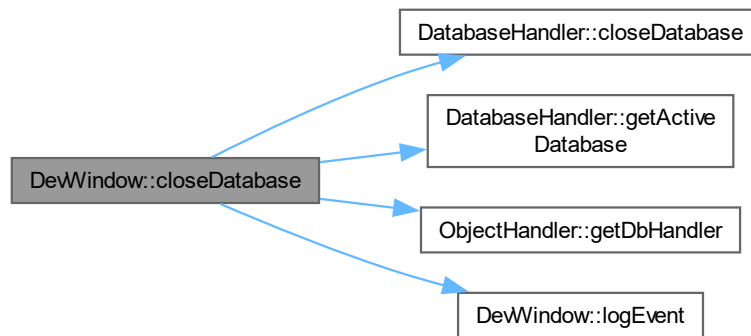


7.2.3.5 closeDatabase()

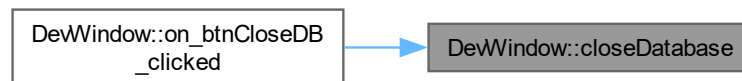
```
void DevWindow::closeDatabase (
    DatabaseHandler * db ) [private]
```

Definition at line 82 of file [devwindow.cpp](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.2.3.6 enableButtons()

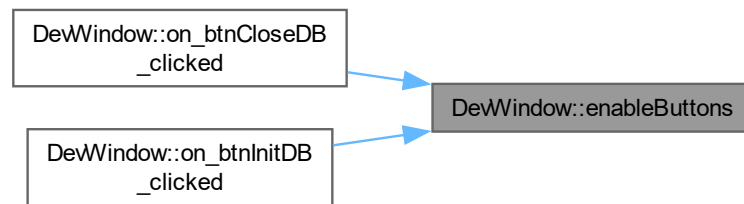
```
void DevWindow::enableButtons (
    bool databaseLoaded ) [private]
```

Definition at line 37 of file [devwindow.cpp](#).

Here is the call graph for this function:



Here is the caller graph for this function:

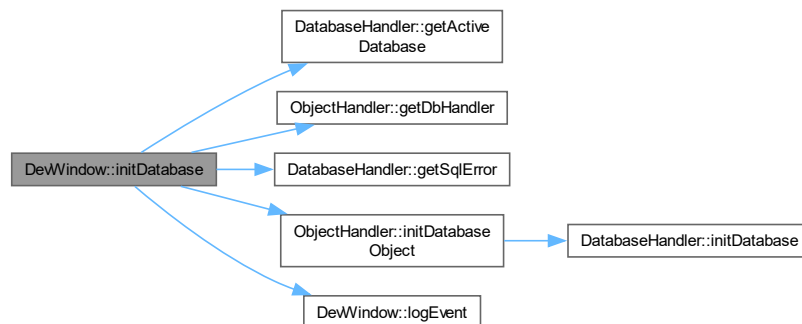


7.2.3.7 initDatabase()

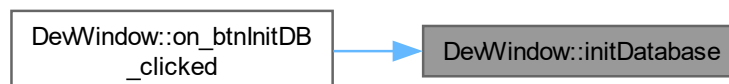
```
void DevWindow::initDatabase (
    ObjectHandler * oH ) [private]
```

Definition at line 71 of file [devwindow.cpp](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.2.3.8 logEvent() [1/3]

```
void DevWindow::logEvent (
    QString message )
```

Definition at line 31 of file [devwindow.cpp](#).

7.2.3.9 logEvent() [2/3]

```
void DevWindow::logEvent (
    QString message,
    QSqlError error )
```

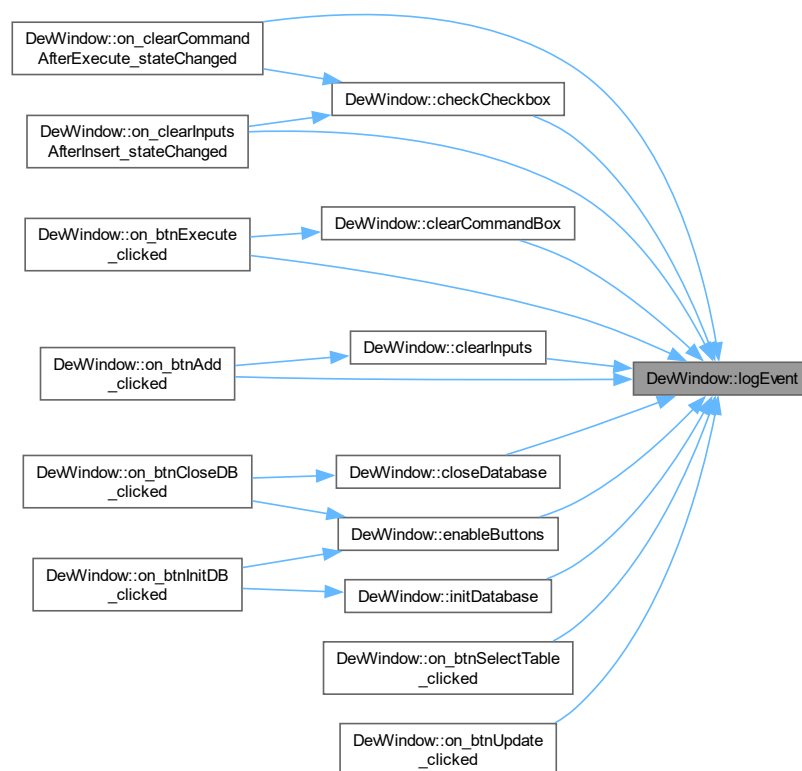
Definition at line 25 of file [devwindow.cpp](#).

7.2.3.10 logEvent() [3/3]

```
void DevWindow::logEvent (
    QString type,
    QString message )
```

Definition at line 19 of file [devwindow.cpp](#).

Here is the caller graph for this function:

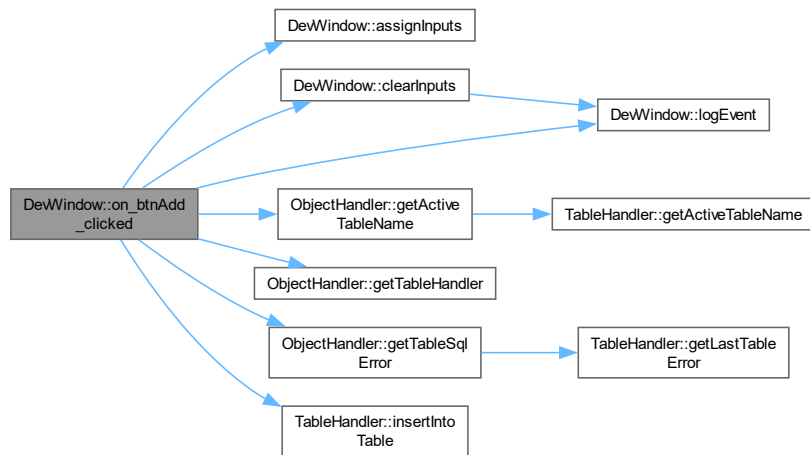


7.2.3.11 on_btnAdd_clicked

```
void DevWindow::on_btnAdd_clicked ( ) [private], [slot]
```

Definition at line 159 of file [devwindow.cpp](#).

Here is the call graph for this function:

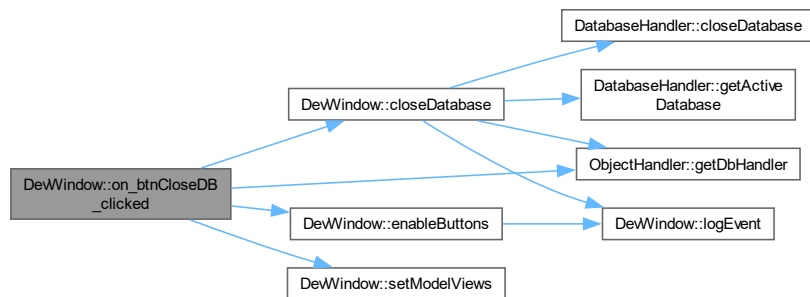


7.2.3.12 on_btnCloseDB_clicked

```
void DevWindow::on_btnCloseDB_clicked ( ) [private], [slot]
```

Definition at line 134 of file `devwindow.cpp`.

Here is the call graph for this function:

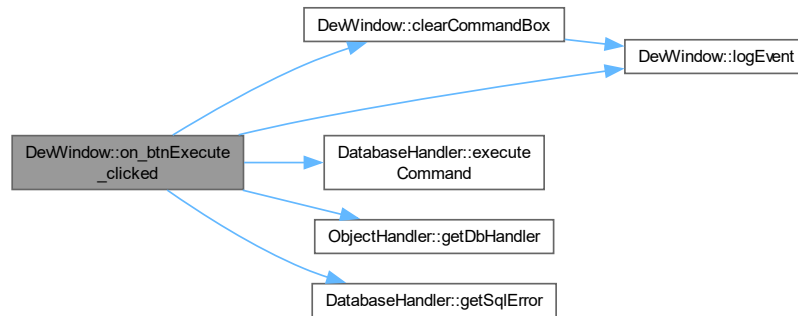


7.2.3.13 on_btnExecute_clicked

```
void DevWindow::on_btnExecute_clicked ( ) [private], [slot]
```

Definition at line 141 of file `devwindow.cpp`.

Here is the call graph for this function:

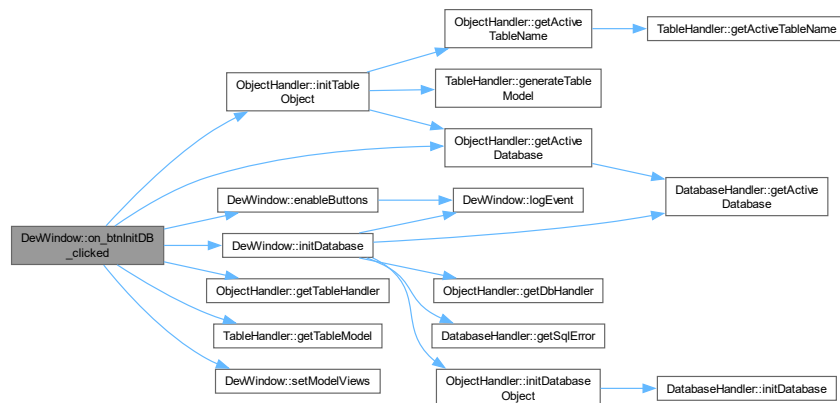


7.2.3.14 on_btnInitDB_clicked

```
void DevWindow::on_btnInitDB_clicked ( ) [private], [slot]
```

Definition at line 123 of file `devwindow.cpp`.

Here is the call graph for this function:

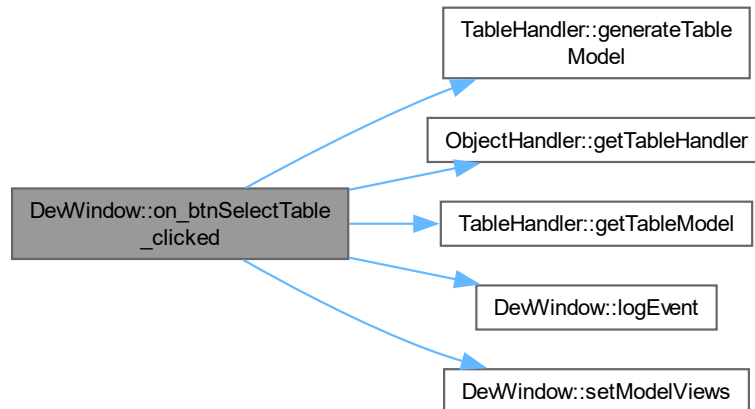


7.2.3.15 on_btnSelectTable_clicked

```
void DevWindow::on_btnSelectTable_clicked ( ) [private], [slot]
```

Definition at line 152 of file `devwindow.cpp`.

Here is the call graph for this function:

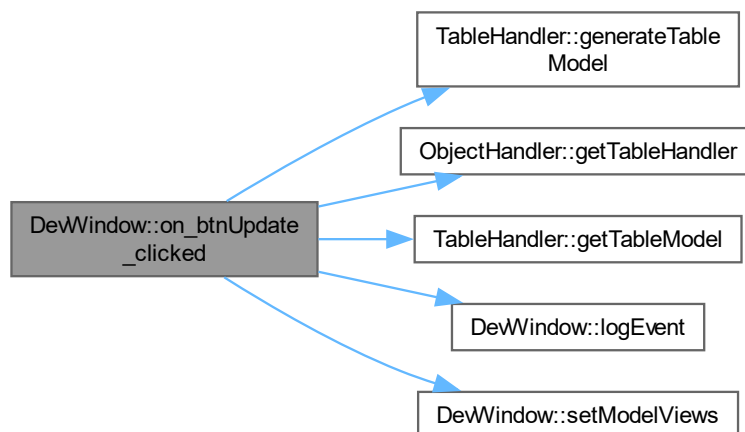


7.2.3.16 on_btnUpdate_clicked

```
void DevWindow::on_btnUpdate_clicked ( ) [private], [slot]
```

Definition at line 171 of file [devwindow.cpp](#).

Here is the call graph for this function:



7.2.3.17 on_clearCommandAfterExecute_stateChanged

```
void DevWindow::on_clearCommandAfterExecute_stateChanged (
    int arg1 ) [private], [slot]
```

Definition at line 179 of file [devwindow.cpp](#).

Here is the call graph for this function:

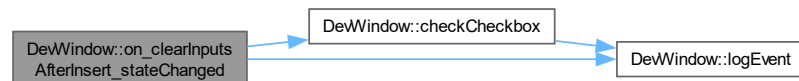


7.2.3.18 on_clearInputsAfterInsert_stateChanged

```
void DevWindow::on_clearInputsAfterInsert_stateChanged (
    int arg1 ) [private], [slot]
```

Definition at line 185 of file [devwindow.cpp](#).

Here is the call graph for this function:

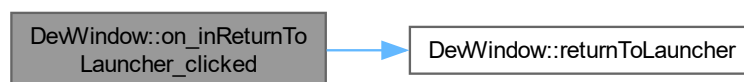


7.2.3.19 on_inReturnToLauncher_clicked

```
void DevWindow::on_inReturnToLauncher_clicked ( ) [private], [slot]
```

Definition at line 191 of file [devwindow.cpp](#).

Here is the call graph for this function:

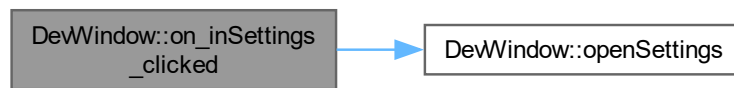


7.2.3.20 on_inSettings_clicked

```
void DevWindow::on_inSettings_clicked ( ) [private], [slot]
```

Definition at line 201 of file [devwindow.cpp](#).

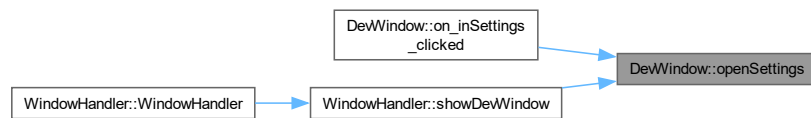
Here is the call graph for this function:



7.2.3.21 openSettings

```
void DevWindow::openSettings ( ) [signal]
```

Here is the caller graph for this function:



7.2.3.22 retranslateUi()

```
void DevWindow::retranslateUi ( ) [virtual]
```

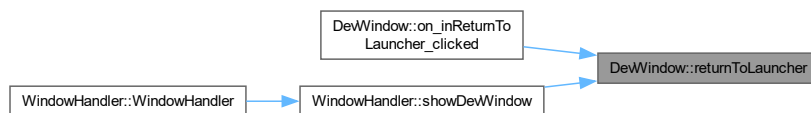
Implements [TranslatableWindow](#).

Definition at line 196 of file [devwindow.cpp](#).

7.2.3.23 returnToLauncher

```
void DevWindow::returnToLauncher ( ) [signal]
```

Here is the caller graph for this function:

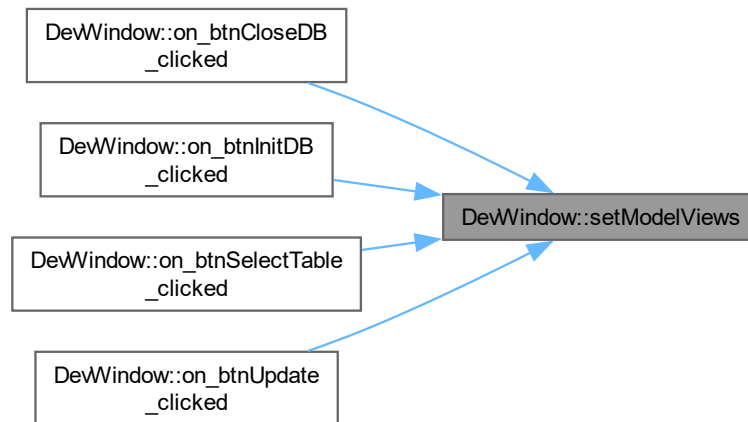


7.2.3.24 setModelViews() [1/2]

```
void DevWindow::setModelViews ( ) [private]
```

Definition at line 55 of file [devwindow.cpp](#).

Here is the caller graph for this function:



7.2.3.25 setModelViews() [2/2]

```
void DevWindow::setModelViews (
    QSqlTableModel & m ) [private]
```

Definition at line 48 of file [devwindow.cpp](#).

7.2.4 Member Data Documentation

7.2.4.1 clearCommand

```
bool DevWindow::clearCommand = false [private]
```

Definition at line 98 of file [devwindow.hpp](#).

7.2.4.2 clearInput

```
bool DevWindow::clearInput = false [private]
```

Definition at line 98 of file [devwindow.hpp](#).

7.2.4.3 input1

```
QString DevWindow::input1 [private]
```

Definition at line 97 of file [devwindow.hpp](#).

7.2.4.4 input2

```
QString DevWindow::input2 [private]
```

Definition at line 97 of file [devwindow.hpp](#).

7.2.4.5 input3

```
QString DevWindow::input3 [private]
```

Definition at line 97 of file [devwindow.hpp](#).

7.2.4.6 input4

QString DevWindow::input4 [private]
Definition at line 97 of file [devwindow.hpp](#).

7.2.4.7 input5

QString DevWindow::input5 [private]
Definition at line 97 of file [devwindow.hpp](#).

7.2.4.8 objectHandler

ObjectHandler* DevWindow::objectHandler [private]
Definition at line 95 of file [devwindow.hpp](#).

7.2.4.9 ui

Ui::DevWindow* DevWindow::ui [private]
Definition at line 94 of file [devwindow.hpp](#).

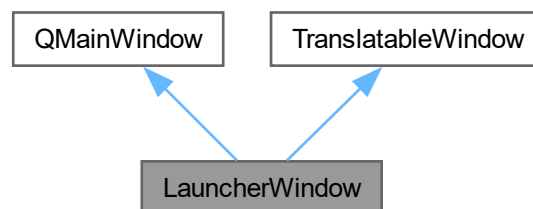
The documentation for this class was generated from the following files:

- C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/[devwindow.hpp](#)
- C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/[devwindow.cpp](#)

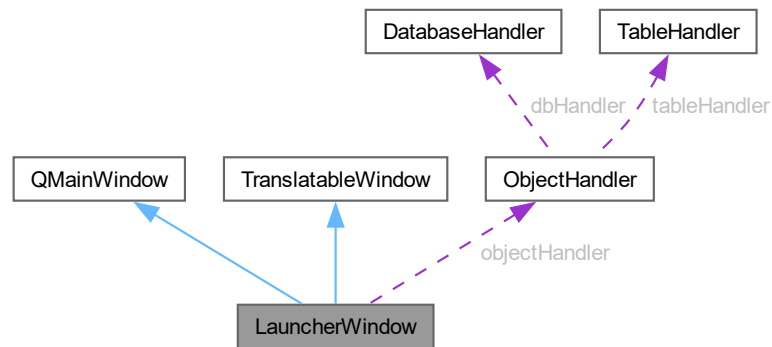
7.3 LauncherWindow Class Reference

```
#include <launcherwindow.hpp>
```

Inheritance diagram for LauncherWindow:



Collaboration diagram for LauncherWindow:



Signals

- void [openDevWindow](#) ()
- void [openSettings](#) ()

Public Member Functions

- [LauncherWindow](#) (QWidget *parent=nullptr, [ObjectHandler](#) *newObjectHandler=nullptr)
- [~LauncherWindow](#) ()
- void [retranslateUi](#) ()

Public Member Functions inherited from [TranslatableWindow](#)

- [TranslatableWindow](#) ()

Public Attributes

- Ui::LauncherWindow * [ui](#)

Private Slots

- void [on_inShowDev_clicked](#) ()
- void [on_pushButton_clicked](#) ()

Private Attributes

- [ObjectHandler](#) * [objectHandler](#)

7.3.1 Detailed Description

Definition at line 12 of file [launcherwindow.hpp](#).

7.3.2 Constructor & Destructor Documentation

7.3.2.1 LauncherWindow()

```

LauncherWindow::LauncherWindow (
    QWidget * parent = nullptr,
    ObjectHandler * newObjectHandler = nullptr ) [explicit]

```

Definition at line 4 of file [launcherwindow.cpp](#).

7.3.2.2 ~LauncherWindow()

LauncherWindow::~~LauncherWindow ()

Definition at line 13 of file [launcherwindow.cpp](#).

7.3.3 Member Function Documentation

7.3.3.1 on_inShowDev_clicked

void LauncherWindow::on_inShowDev_clicked () [private], [slot]

Definition at line 19 of file [launcherwindow.cpp](#).

Here is the call graph for this function:

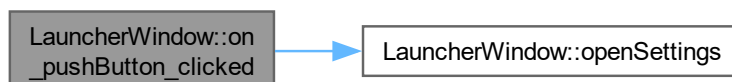


7.3.3.2 on_pushButton_clicked

void LauncherWindow::on_pushButton_clicked () [private], [slot]

Definition at line 27 of file [launcherwindow.cpp](#).

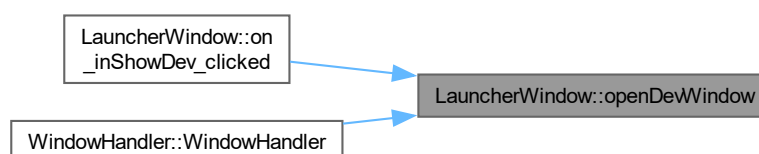
Here is the call graph for this function:



7.3.3.3 openDevWindow

void LauncherWindow::openDevWindow () [signal]

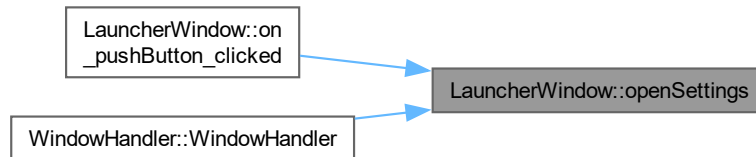
Here is the caller graph for this function:



7.3.3.4 openSettings

```
void LauncherWindow::openSettings ( ) [signal]
```

Here is the caller graph for this function:



7.3.3.5 retranslateUi()

```
void LauncherWindow::retranslateUi ( ) [virtual]
```

Implements [TranslatableWindow](#).

Definition at line 32 of file [launcherwindow.cpp](#).

7.3.4 Member Data Documentation

7.3.4.1 objectHandler

```
ObjectHandler* LauncherWindow::objectHandler [private]
```

Definition at line 31 of file [launcherwindow.hpp](#).

7.3.4.2 ui

```
Ui::LauncherWindow* LauncherWindow::ui
```

Definition at line 20 of file [launcherwindow.hpp](#).

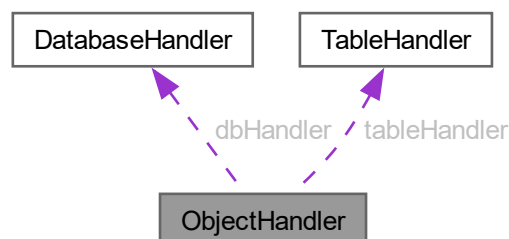
The documentation for this class was generated from the following files:

- C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/[launcherwindow.hpp](#)
- C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/[launcherwindow.cpp](#)

7.4 ObjectHandler Class Reference

```
#include <objecthandler.hpp>
```

Collaboration diagram for ObjectHandler:



Public Member Functions

- [ObjectHandler](#) (QApplication *newApplication, [DatabaseHandler](#) *newDBHandler, [TableHandler](#) *newTableHandler)
- void [setActiveTableName](#) (const QString &newActiveTableName)
- bool [initDatabaseObject](#) ()
- bool [initTableObject](#) ()
- bool [initTableObject](#) (QSqlDatabase inputActiveDatabase, const QString &inputTableName)
- const QSqlDatabase & [getActiveDatabase](#) () const
- [TableHandler](#) * [getTableHandler](#) () const
- [DatabaseHandler](#) * [getDbHandler](#) () const
- const QString & [getActiveTableName](#) () const
- const QSqlError & [getTableSqlError](#) () const

Public Attributes

- QApplication * [application](#)

Private Attributes

- [DatabaseHandler](#) * [dbHandler](#)
- [TableHandler](#) * [tableHandler](#)

7.4.1 Detailed Description

Definition at line 11 of file [objecthandler.hpp](#).

7.4.2 Constructor & Destructor Documentation**7.4.2.1 ObjectHandler()**

```
ObjectHandler::ObjectHandler (
    QApplication * newApplication,
    DatabaseHandler * newDBHandler,
    TableHandler * newTableHandler )
```

Definition at line 3 of file [objecthandler.cpp](#).

7.4.3 Member Function Documentation**7.4.3.1 getActiveDatabase()**

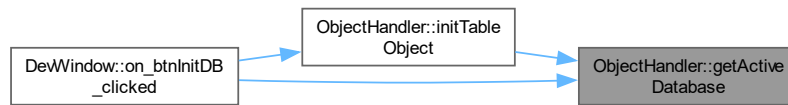
```
const QSqlDatabase & ObjectHandler::getActiveDatabase ( ) const
```

Definition at line 54 of file [objecthandler.cpp](#).

Here is the call graph for this function:



Here is the caller graph for this function:

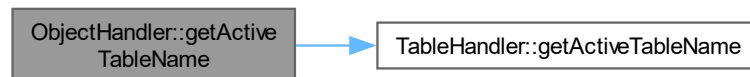


7.4.3.2 `getActiveTableName()`

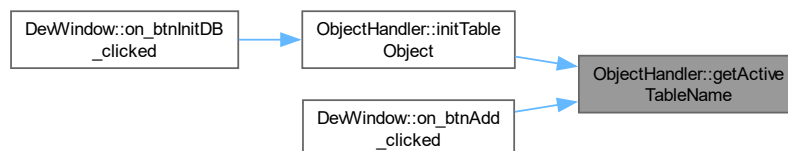
```
const QString & ObjectHandler::getActiveTableName ( ) const
```

Definition at line 43 of file [objecthandler.cpp](#).

Here is the call graph for this function:



Here is the caller graph for this function:

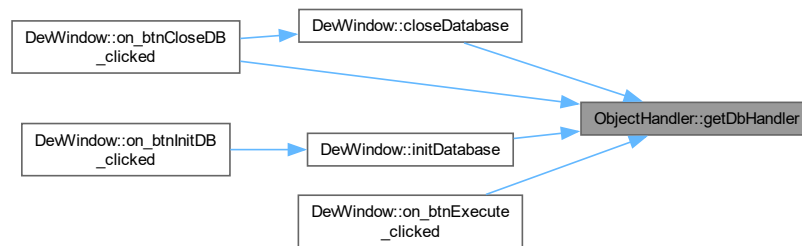


7.4.3.3 `getDbHandler()`

```
DatabaseHandler * ObjectHandler::getDbHandler ( ) const
```

Definition at line 38 of file [objecthandler.cpp](#).

Here is the caller graph for this function:

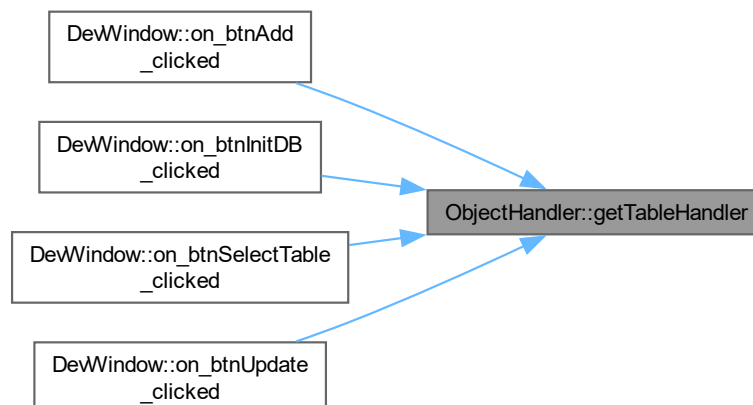


7.4.3.4 getTableHandler()

`TableHandler * ObjectHandler::getTableHandler () const`

Definition at line 33 of file [objecthandler.cpp](#).

Here is the caller graph for this function:



7.4.3.5 getTableSqlError()

`const QSqlError & ObjectHandler::getTableSqlError () const`

Definition at line 48 of file [objecthandler.cpp](#).

Here is the call graph for this function:



Here is the caller graph for this function:

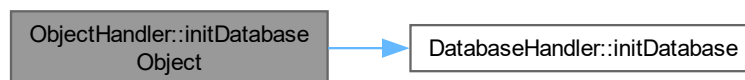


7.4.3.6 initDatabaseObject()

```
bool ObjectHandler::initDatabaseObject ( )
```

Definition at line 10 of file [objecthandler.cpp](#).

Here is the call graph for this function:



Here is the caller graph for this function:

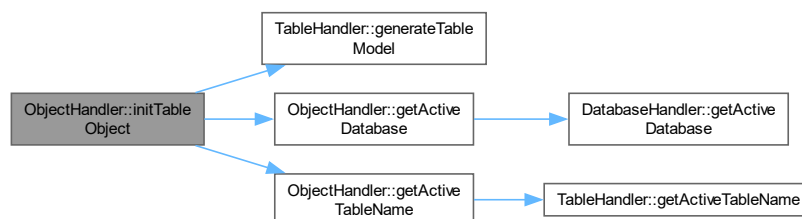


7.4.3.7 initTableObject() [1/2]

```
bool ObjectHandler::initTableObject ( )
```

Definition at line 19 of file [objecthandler.cpp](#).

Here is the call graph for this function:



Here is the caller graph for this function:

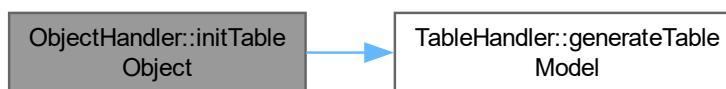


7.4.3.8 initTableObject() [2/2]

```
bool ObjectHandler::initTableObject (
    QSqlDatabase inputActiveDatabase,
    const QString & inputTableName )
```

Definition at line 27 of file [objecthandler.cpp](#).

Here is the call graph for this function:

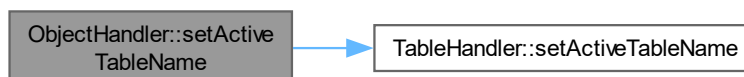


7.4.3.9 setActiveTableName()

```
void ObjectHandler::setActiveTableName (
    const QString & newActiveTableName )
```

Definition at line 59 of file [objecthandler.cpp](#).

Here is the call graph for this function:



7.4.4 Member Data Documentation

7.4.4.1 application

```
QApplication* ObjectHandler::application
```

Definition at line 41 of file [objecthandler.hpp](#).

7.4.4.2 dbHandler

`DatabaseHandler* ObjectHandler::dbHandler [private]`

Definition at line 48 of file [objecthandler.hpp](#).

7.4.4.3 tableHandler

`TableHandler* ObjectHandler::tableHandler [private]`

Definition at line 49 of file [objecthandler.hpp](#).

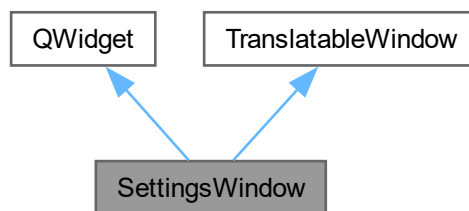
The documentation for this class was generated from the following files:

- C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/backendClasses/[objecthandler.hpp](#)
- C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/backendClasses/[objecthandler.cpp](#)

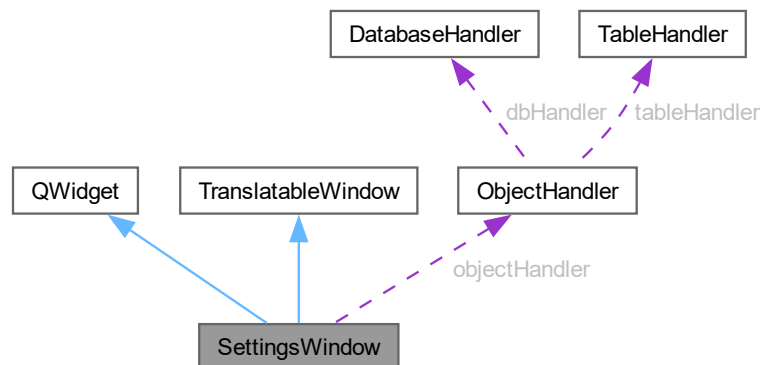
7.5 SettingsWindow Class Reference

`#include <settingswindow.hpp>`

Inheritance diagram for SettingsWindow:



Collaboration diagram for SettingsWindow:



Signals

- void [appliedSettings](#) ()

Public Member Functions

- [SettingsWindow](#) (QWidget *parent=nullptr, [ObjectHandler](#) *objectHandler=nullptr)
- [~SettingsWindow](#) ()
- void [retranslateUi](#) ()

Public Member Functions inherited from [TranslatableWindow](#)

- [TranslatableWindow](#) ()

Private Slots

- void [on_inClose_clicked](#) ()
- void [on_inApply_clicked](#) ()
- void [on_inLanguage_currentIndexChanged](#) (int index)

Private Member Functions

- void [installTranslator](#) ()

Private Attributes

- Ui::SettingsWindow * [ui](#)
- QString [language](#)
- [ObjectHandler](#) * [objectHandler](#)
- QTranslator * [translator](#)
- int [languageIndex](#)
- int [tempLanguageIndex](#)
- bool [languageChanged](#)

7.5.1 Detailed Description

Definition at line 14 of file [settingswindow.hpp](#).

7.5.2 Constructor & Destructor Documentation**7.5.2.1 SettingsWindow()**

```
SettingsWindow::SettingsWindow (
    QWidget * parent = nullptr,
    ObjectHandler * objectHandler = nullptr ) [explicit]
```

Definition at line 4 of file [settingswindow.cpp](#).

7.5.2.2 ~SettingsWindow()

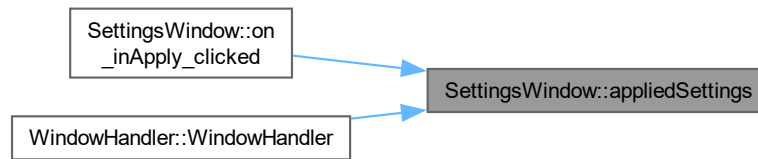
```
SettingsWindow::~SettingsWindow ( )
```

Definition at line 13 of file [settingswindow.cpp](#).

7.5.3 Member Function Documentation**7.5.3.1 appliedSettings**

```
void SettingsWindow::appliedSettings ( ) [signal]
```

Here is the caller graph for this function:



7.5.3.2 installTranslator()

```
void SettingsWindow::installTranslator ( ) [private]
```

Definition at line 54 of file [settingswindow.cpp](#).

Here is the caller graph for this function:

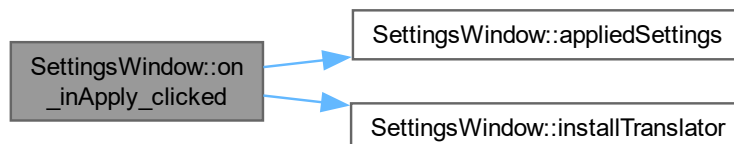


7.5.3.3 on_inApply_clicked

```
void SettingsWindow::on_inApply_clicked ( ) [private], [slot]
```

Definition at line 29 of file [settingswindow.cpp](#).

Here is the call graph for this function:



7.5.3.4 on_inClose_clicked

```
void SettingsWindow::on_inClose_clicked ( ) [private], [slot]
```

Definition at line 23 of file [settingswindow.cpp](#).

7.5.3.5 on_inLanguage_currentIndexChanged

```
void SettingsWindow::on_inLanguage_currentIndexChanged (
    int index ) [private], [slot]
```

Definition at line 41 of file [settingswindow.cpp](#).

7.5.3.6 retranslateUi()

```
void SettingsWindow::retranslateUi ( ) [virtual]
```

Implements [TranslatableWindow](#).

Definition at line 18 of file [settingswindow.cpp](#).

7.5.4 Member Data Documentation

7.5.4.1 language

```
QString SettingsWindow::language [private]
```

Definition at line 37 of file [settingswindow.hpp](#).

7.5.4.2 languageChanged

```
bool SettingsWindow::languageChanged [private]
```

Definition at line 41 of file [settingswindow.hpp](#).

7.5.4.3 languageIndex

```
int SettingsWindow::languageIndex [private]
```

Definition at line 40 of file [settingswindow.hpp](#).

7.5.4.4 objectHandler

```
ObjectHandler* SettingsWindow::objectHandler [private]
```

Definition at line 38 of file [settingswindow.hpp](#).

7.5.4.5 tempLanguageIndex

```
int SettingsWindow::tempLanguageIndex [private]
```

Definition at line 40 of file [settingswindow.hpp](#).

7.5.4.6 translator

```
QTranslator* SettingsWindow::translator [private]
```

Definition at line 39 of file [settingswindow.hpp](#).

7.5.4.7 ui

```
Ui::SettingsWindow* SettingsWindow::ui [private]
```

Definition at line 36 of file [settingswindow.hpp](#).

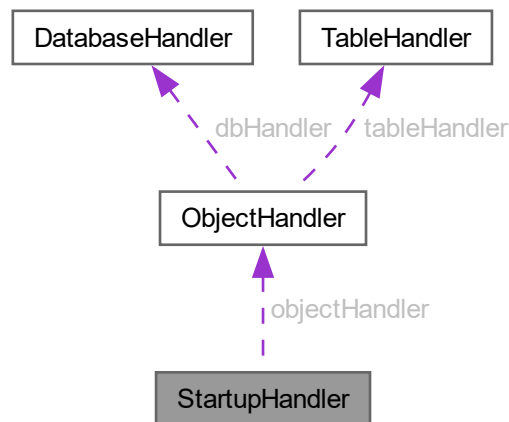
The documentation for this class was generated from the following files:

- C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/[settingswindow.hpp](#)
- C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/[settingswindow.cpp](#)

7.6 StartupHandler Class Reference

```
#include <startuphandler.hpp>
```

Collaboration diagram for StartupHandler:



Public Member Functions

- [StartupHandler](#) (QApplication *application)
- [~StartupHandler](#) ()
- [ObjectHandler * startUp](#) ()

Private Member Functions

- void [initTranslator](#) ()
- void [initObjectHandler](#) ()

Private Attributes

- QApplication * [ptrApplication](#)
- [ObjectHandler * objectHandler](#)

7.6.1 Detailed Description

Definition at line 11 of file [startuphandler.hpp](#).

7.6.2 Constructor & Destructor Documentation

7.6.2.1 StartupHandler()

```
StartupHandler::StartupHandler (
    QApplication * application )
```

Definition at line 3 of file [startuphandler.cpp](#).

7.6.2.2 ~StartupHandler()

```
StartupHandler::~StartupHandler ( )
```

Definition at line 8 of file [startuphandler.cpp](#).

Here is the caller graph for this function:



7.6.3 Member Function Documentation

7.6.3.1 initObjectHandler()

```
void StartupHandler::initObjectHandler ( ) [private]
```

Definition at line 35 of file [startuphandler.cpp](#).

Here is the caller graph for this function:



7.6.3.2 initTranslator()

```
void StartupHandler::initTranslator ( ) [private]
```

Definition at line 22 of file [startuphandler.cpp](#).

Here is the caller graph for this function:

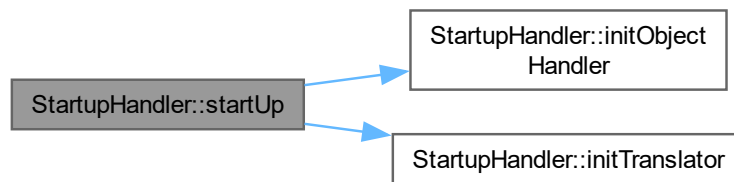


7.6.3.3 startUp()

```
ObjectHandler * StartupHandler::startUp ( )
```

Definition at line 13 of file [startuphandler.cpp](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.4 Member Data Documentation

7.6.4.1 objectHandler

`ObjectHandler*` `StartupHandler::objectHandler` [private]

The `ObjectHandler` instance.

Definition at line 35 of file [startuphandler.hpp](#).

7.6.4.2 ptrApplication

`QApplication*` `StartupHandler::ptrApplication` [private]

The `QApplication` instance.

Definition at line 33 of file [startuphandler.hpp](#).

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

- [C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/backendClasses/startuphandler.hpp](#)
- [C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/backendClasses/startuphandler.cpp](#)

7.7 TableHandler Class Reference

```
#include <tablehandler.hpp>
```

Public Member Functions

- [TableHandler](#) ()
- [TableHandler](#) (const QSqlDatabase &newActiveDatabase, const QString &tableName)
- void [generateTableModel](#) ()
- void [generateTableModel](#) (QSqlDatabase &activeDatabase, const QString &tableName)
- QSqlTableModel * [getTableModel](#) () const
- const QSqlError & [getTableSQLError](#) () const

- void [setTableName](#) (const QString &newTableName)
- void [setActiveDatabase](#) (QSqlDatabase newActiveDatabase)
- bool [insertIntoTable](#) (const QString &tableName, const QString &value1, const QString &value2, const QString &value3, const QString &value4, const QString &value5)
- const QString & [getActiveTableName](#) () const
- void [setActiveTableName](#) (const QString &newActiveTableName)
- const QSqlDatabase & [getTblActiveDatabase](#) () const
- const QSqlError & [getLastTableError](#) () const

Private Attributes

- QString [activeTableName](#) = "userTable"
- QSqlTableModel * [tableModel](#)
- QSqlDatabase [tblActiveDatabase](#)
- QSqlError [lastTableError](#)

7.7.1 Detailed Description

Definition at line 7 of file [tablehandler.hpp](#).

7.7.2 Constructor & Destructor Documentation

7.7.2.1 TableHandler() [1/2]

```
TableHandler::TableHandler ( )
```

Definition at line 3 of file [tablehandler.cpp](#).

7.7.2.2 TableHandler() [2/2]

```
TableHandler::TableHandler (
    const QSqlDatabase & newActiveDatabase,
    const QString & tableName )
```

Definition at line 8 of file [tablehandler.cpp](#).

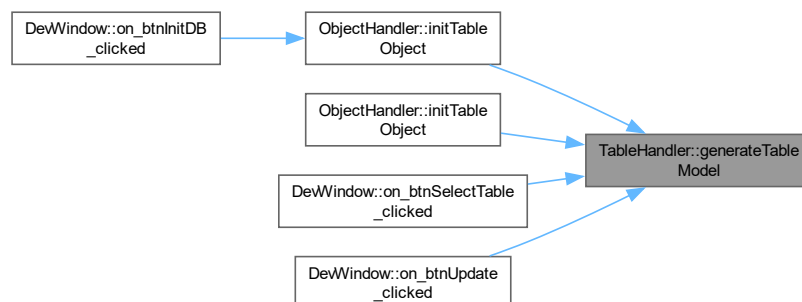
7.7.3 Member Function Documentation

7.7.3.1 generateTableModel() [1/2]

```
void TableHandler::generateTableModel ( )
```

Definition at line 17 of file [tablehandler.cpp](#).

Here is the caller graph for this function:



7.7.3.2 generateTableModel() [2/2]

```
void TableHandler::generateTableModel (
    QSqlDatabase & activeDatabase,
    const QString & tableName )
```

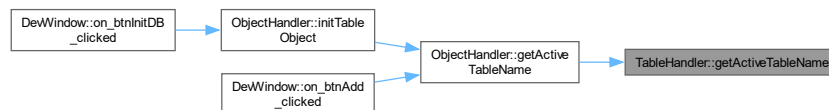
Definition at line 24 of file [tablehandler.cpp](#).

7.7.3.3 getActiveTableName()

```
const QString & TableHandler::getActiveTableName ( ) const
```

Definition at line 50 of file [tablehandler.cpp](#).

Here is the caller graph for this function:

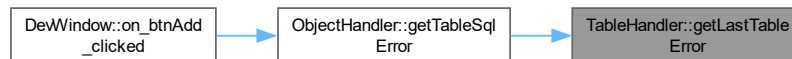


7.7.3.4 getLastTableError()

```
const QSqlError & TableHandler::getLastTableError ( ) const
```

Definition at line 65 of file [tablehandler.cpp](#).

Here is the caller graph for this function:

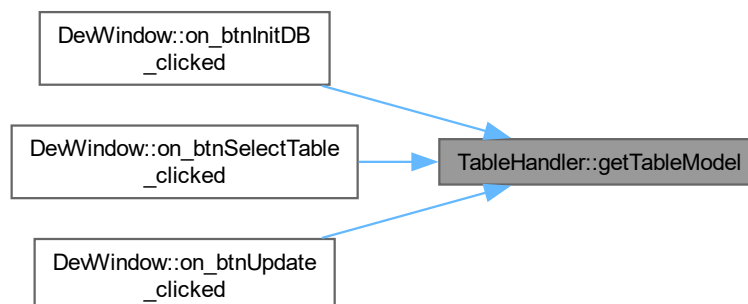


7.7.3.5 getTableModel()

```
QSqlTableModel * TableHandler::getTableModel ( ) const
```

Definition at line 70 of file [tablehandler.cpp](#).

Here is the caller graph for this function:



7.7.3.6 getTableSQLError()

```
const QSqlError & TableHandler::getTableSQLError ( ) const
```

7.7.3.7 getTblActiveDatabase()

```
const QSqlDatabase & TableHandler::getTblActiveDatabase ( ) const
```

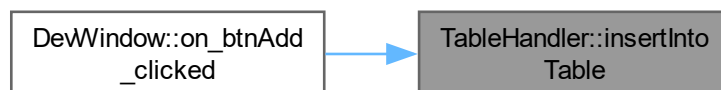
Definition at line 60 of file [tablehandler.cpp](#).

7.7.3.8 insertIntoTable()

```
bool TableHandler::insertIntoTable (
    const QString & tableName,
    const QString & value1,
    const QString & value2,
    const QString & value3,
    const QString & value4,
    const QString & value5 )
```

Definition at line 33 of file [tablehandler.cpp](#).

Here is the caller graph for this function:

**7.7.3.9 setActiveDatabase()**

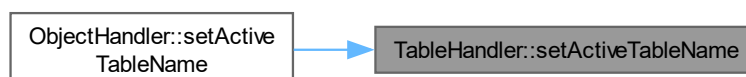
```
void TableHandler::setActiveDatabase (
    QSqlDatabase newActiveDatabase )
```

7.7.3.10 setActiveTableName()

```
void TableHandler::setActiveTableName (
    const QString & newActiveTableName )
```

Definition at line 55 of file [tablehandler.cpp](#).

Here is the caller graph for this function:

**7.7.3.11 setTableName()**

```
void TableHandler::setTableName (
    const QString & newTableName )
```

7.7.4 Member Data Documentation

7.7.4.1 activeTableName

`QString TableHandler::activeTableName = "userTable" [private]`

Definition at line 53 of file [tablehandler.hpp](#).

7.7.4.2 lastTableError

`QSqlError TableHandler::lastTableError [private]`

Definition at line 59 of file [tablehandler.hpp](#).

7.7.4.3 tableModel

`QSqlTableModel* TableHandler::tableModel [private]`

Definition at line 55 of file [tablehandler.hpp](#).

7.7.4.4 tblActiveDatabase

`QSqlDatabase TableHandler::tblActiveDatabase [private]`

Definition at line 57 of file [tablehandler.hpp](#).

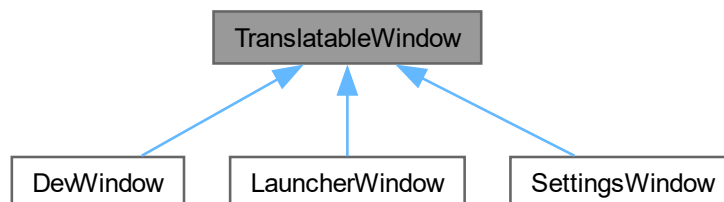
The documentation for this class was generated from the following files:

- C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/databaseClasses/[tablehandler.hpp](#)
- C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/databaseClasses/[tablehandler.cpp](#)

7.8 TranslatableWindow Class Reference

`#include <translatablewindow.hpp>`

Inheritance diagram for TranslatableWindow:



Public Member Functions

- [TranslatableWindow](#) ()
- virtual void [retranslateUi](#) ()=0

7.8.1 Detailed Description

Definition at line 4 of file [translatablewindow.hpp](#).

7.8.2 Constructor & Destructor Documentation

7.8.2.1 TranslatableWindow()

`TranslatableWindow::TranslatableWindow ()`

Definition at line 3 of file [translatablewindow.cpp](#).

7.8.3 Member Function Documentation

7.8.3.1 retranslateUi()

`virtual void TranslatableWindow::retranslateUi () [pure virtual]`

Implemented in [DevWindow](#), [LauncherWindow](#), and [SettingsWindow](#).

Here is the caller graph for this function:



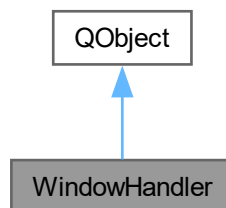
The documentation for this class was generated from the following files:

- C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/[translatablewindow.hpp](#)
- C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/[translatablewindow.cpp](#)

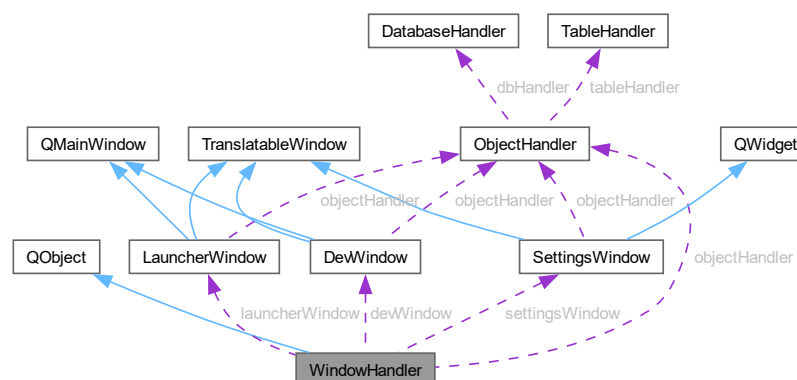
7.9 WindowHandler Class Reference

```
#include <windowhandler.hpp>
```

Inheritance diagram for WindowHandler:



Collaboration diagram for WindowHandler:



Here is the call graph for this function:



Here is the caller graph for this function:

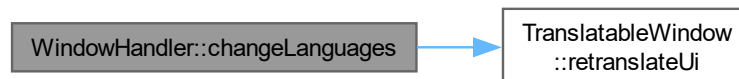


7.9.3.2 changeLanguages()

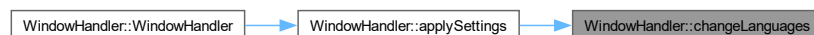
```
void WindowHandler::changeLanguages ( ) [private]
```

Definition at line 49 of file [windowhandler.cpp](#).

Here is the call graph for this function:



Here is the caller graph for this function:

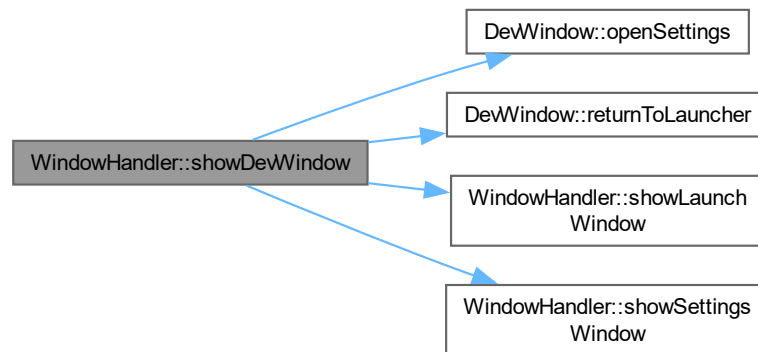


7.9.3.3 showDevWindow

```
void WindowHandler::showDevWindow ( ) [private], [slot]
```

Definition at line 27 of file [windowhandler.cpp](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.9.3.4 showLaunchWindow

```
void WindowHandler::showLaunchWindow ( ) [slot]
```

Definition at line 17 of file [windowhandler.cpp](#).

Here is the caller graph for this function:

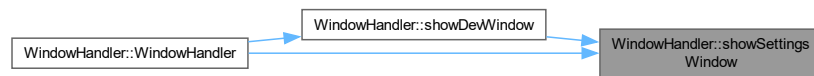


7.9.3.5 showSettingsWindow

```
void WindowHandler::showSettingsWindow ( ) [private], [slot]
```

Definition at line 39 of file [windowhandler.cpp](#).

Here is the caller graph for this function:



7.9.4 Member Data Documentation

7.9.4.1 devWindow

`DevWindow*` `WindowHandler::devWindow = nullptr` [private]

Definition at line 29 of file [windowhandler.hpp](#).

7.9.4.2 launcherWindow

`LauncherWindow*` `WindowHandler::launcherWindow = nullptr` [private]

Definition at line 28 of file [windowhandler.hpp](#).

7.9.4.3 objectHandler

`ObjectHandler*` `WindowHandler::objectHandler` [private]

Definition at line 31 of file [windowhandler.hpp](#).

7.9.4.4 settingsWindow

`SettingsWindow*` `WindowHandler::settingsWindow = nullptr` [private]

Definition at line 30 of file [windowhandler.hpp](#).

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

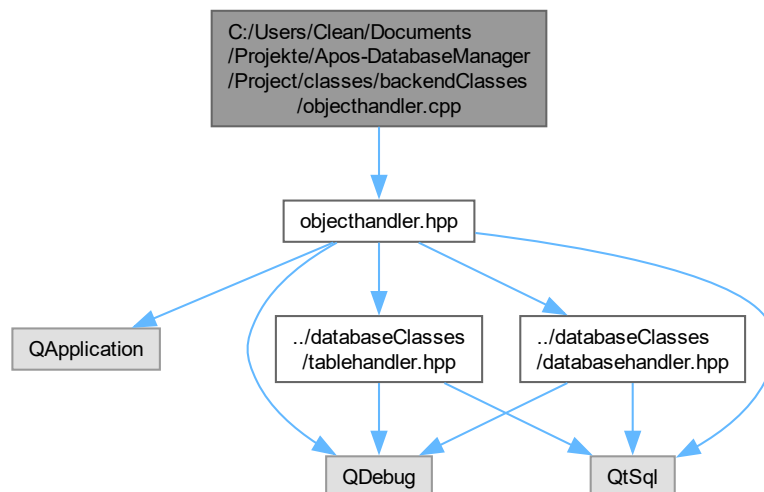
- `C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/windowhandler.hpp`
- `C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/windowhandler.cpp`

Chapter 8

File Documentation

8.1 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/↵ Project/classes/backendClasses/objecthandler.cpp File Reference

#include "objecthandler.hpp"
Include dependency graph for objecthandler.cpp:



8.2 objecthandler.cpp

[Go to the documentation of this file.](#)

```
00001 #include "objecthandler.hpp"
00002
00003 ObjectHandler::ObjectHandler(QApplication* newApplication, DatabaseHandler* newDBHandler,
00004                             TableHandler* newTableHandler)
00005 {
00006     application = newApplication;
00007     dbHandler = newDBHandler;
00008     tableHandler = newTableHandler;
00009 }
00010 bool ObjectHandler::initDatabaseObject()
00011 {
00012     if (!dbHandler->initDatabase())
00013     {
00014         return false;
00015     }
00016 }
```

```

00016     return true;
00017 }
00018
00019 bool ObjectHandler::initTableObject ()
00020 {
00021     QSqlDatabase db = getActiveDatabase();
00022     QString tn = getActiveTableName();
00023     tableHandler->generateTableModel(db, tn);
00024     return true;
00025 }
00026
00027 bool ObjectHandler::initTableObject(QSqlDatabase inputActiveDatabase, const QString &inputTableName)
00028 {
00029     tableHandler->generateTableModel(inputActiveDatabase, inputTableName);
00030     return true;
00031 }
00032
00033 TableHandler *ObjectHandler::getTableHandler() const
00034 {
00035     return tableHandler;
00036 }
00037
00038 DatabaseHandler *ObjectHandler::getDbHandler() const
00039 {
00040     return dbHandler;
00041 }
00042
00043 const QString &ObjectHandler::getActiveTableName() const
00044 {
00045     return tableHandler->getActiveTableName();
00046 }
00047
00048 const QSqlError& ObjectHandler::getTableSqlError() const
00049 {
00050     return tableHandler->getLastTableError();
00051 }
00052
00053
00054 const QSqlDatabase &ObjectHandler::getActiveDatabase() const
00055 {
00056     return dbHandler->getActiveDatabase();
00057 }
00058
00059 void ObjectHandler::setActiveTableName(const QString &newActiveTableName)
00060 {
00061     tableHandler->setActiveTableName(newActiveTableName);
00062 }
00063

```

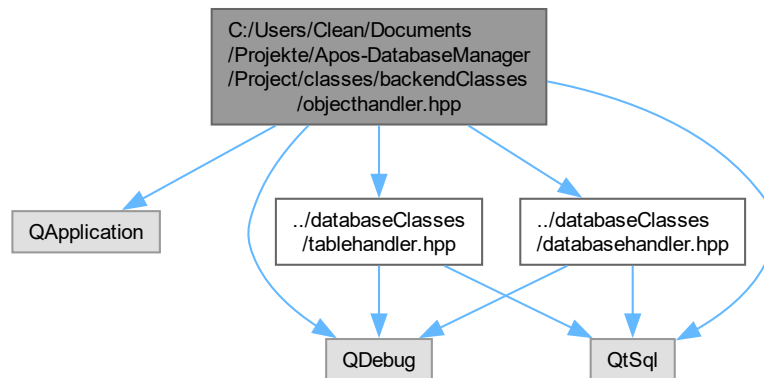
8.3 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/↵ Project/classes/backendClasses/objecthandler.hpp File Reference

```

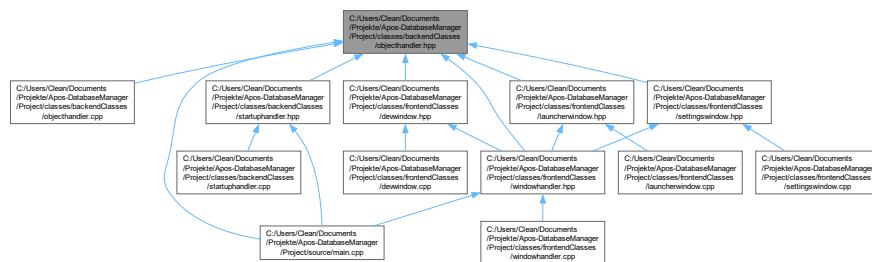
#include <QApplication>
#include <QDebug>
#include <QtSql>
#include "../databaseClasses/databasehandler.hpp"
#include "../databaseClasses/tablehandler.hpp"

```


Include dependency graph for objecthandler.hpp:



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



Classes

- class [ObjectHandler](#)

8.4 objecthandler.hpp

[Go to the documentation of this file.](#)

```

00001 #pragma once
00002
00003 #include <QApplication>
00004 #include <QDebug>
00005 #include <QtSql>
00006
00007 #include "../databaseClasses/databasehandler.hpp"
00008 #include "../databaseClasses/tablehandler.hpp"
00009
00010
00011 class ObjectHandler
00012 {
00013 public:
00014
00015     ObjectHandler(QApplication* newApplication, DatabaseHandler* newDBHandler, TableHandler*
newTableHandler);
00017
00018     void setActiveTableName(const QString &newActiveTableName);
00019
00020     bool initDatabaseObject();
00021
00022
00023

```

```

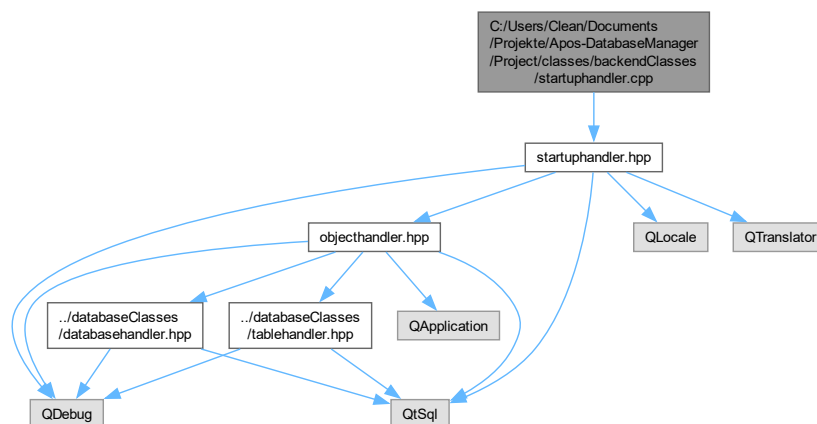
00024     bool initTableObject();
00025
00026
00027     bool initTableObject(QSqlDatabase inputActiveDatabase, const QString &inputTableName);
00028
00029     const QSqlDatabase &getActiveDatabase() const;
00030
00031     TableHandler *getTableHandler() const;
00032
00033
00034     DatabaseHandler *getDbHandler() const;
00035
00036
00037     const QString &getActiveTableName() const;
00038
00039     const QSqlError& getTableSqlError() const;
00040
00041     QApplication *application;
00042
00043
00044
00045 private:
00046
00047
00048     DatabaseHandler *dbHandler;
00049     TableHandler *tableHandler;
00050
00051 };
00052
00053

```

8.5 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/backendClasses/startuphandler.cpp File Reference

```
#include "startuphandler.hpp"
```

Include dependency graph for startuphandler.cpp:



8.6 startuphandler.cpp

[Go to the documentation of this file.](#)

```

00001 #include "startuphandler.hpp"
00002
00003 StartupHandler::StartupHandler(QApplication *application)
00004 {
00005     this->ptrApplication = application;
00006 }
00007
00008 StartupHandler::~StartupHandler()
00009 {
00010     qDebug() << "StartupHandler deleted!";
00011 }

```

```

00012
00013 ObjectHandler* StartupHandler::startUp()
00014 {
00015     initObjectHandler();
00016     initTranslator();
00017     return objectHandler;
00018 }
00019
00020 void StartupHandler::initTranslator()
00021 {
00022     QTranslator translator;
00023     const QStringList uiLanguages = QLocale::system().uiLanguages();
00024     for (const QString &locale : uiLanguages) {
00025         const QString baseName = "Apos-DatabaseManager_" + QLocale(locale).name();
00026         if (translator.load(":/i18n/" + baseName)) {
00027             ptrApplication->installTranslator(&translator);
00028             break;
00029         }
00030     }
00031 }
00032
00033 void StartupHandler::initObjectHandler()
00034 {
00035     DatabaseHandler *dbHandler = new DatabaseHandler();
00036     TableHandler *tableHandler = new TableHandler();
00037     objectHandler = new ObjectHandler(ptrApplication, dbHandler, tableHandler);
00038 }
00039
00040
00041

```

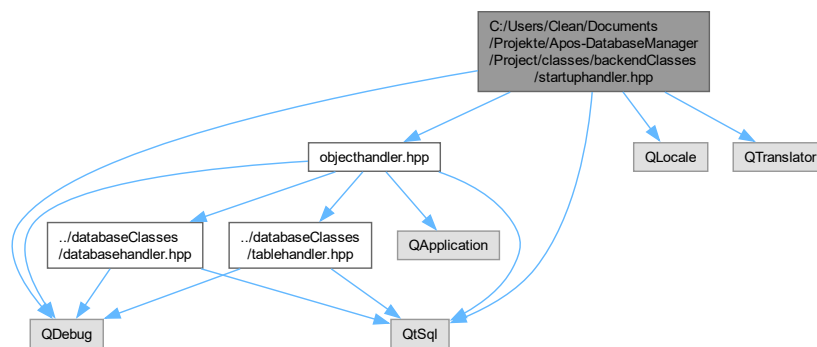
8.7 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/backendClasses/startuphandler.hpp File Reference

```

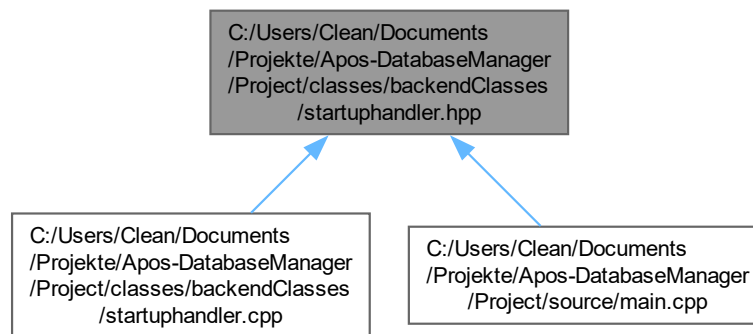
#include <QDebug>
#include <QtSql>
#include <QLocale>
#include <QTranslator>
#include "objecthandler.hpp"

```

Include dependency graph for startuphandler.hpp:



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



Classes

- class [StartupHandler](#)

8.8 startuphandler.hpp

[Go to the documentation of this file.](#)

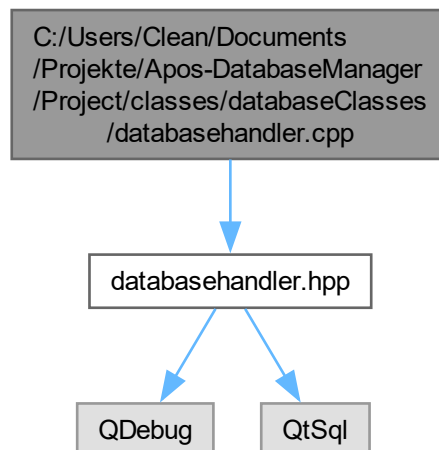
```

00001 #pragma once
00002
00003 #include <QDebug>
00004 #include <QtSql>
00005 #include <QLocale>
00006 #include <QTranslator>
00007
00008 #include "objecthandler.hpp"
00009
00010
00011 class StartupHandler
00012 {
00013 public:
00014
00015     StartupHandler(QApplication* application);
00016
00017     ~StartupHandler();
00018
00019     ObjectHandler* startUp();
00020
00021 private:
00022
00023     void initTranslator();
00024
00025     void initObjectHandler();
00026
00027
00028
00029
00030
00031
00032
00033     QApplication *ptrApplication;
00034     ObjectHandler *objectHandler;
00035 };
00036
00037
  
```

8.9 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/databaseClasses/databasehandler.cpp File Reference

#include "databasehandler.hpp"

Include dependency graph for databasehandler.cpp:



8.10 databasehandler.cpp

[Go to the documentation of this file.](#)

```
00001 #include "databasehandler.hpp"
00002
00003 DatabaseHandler::DatabaseHandler()
00004 {
00005 }
00006 }
00007
00008 bool DatabaseHandler::initDatabase()
00009 {
00010     activeDatabase = QSqlDatabase::addDatabase("QSQLITE", "connection1");
00011     activeDatabase.setDatabaseName(databasePath);
00012
00013     if (!activeDatabase.open())
00014     {
00015         lastSqlError = activeDatabase.lastError();
00016         qDebug() << lastSqlError.text();
00017         return false;
00018     }
00019     else
00020     {
00021         return true;
00022     }
00023 }
00024
00025 void DatabaseHandler::closeDatabase()
00026 {
00027     activeDatabase.close();
00028     activeDatabase = QSqlDatabase();
00029 }
00030
00031 bool DatabaseHandler::executeCommand(QString command)
00032 {
00033     QSqlQuery query(activeDatabase);
00034     if (!query.exec(command))
00035     {
00036         lastSqlError = query.lastError();
```

```

00037         qDebug() << lastSqlError.text();
00038         return false;
00039     }
00040     return true;
00041 }
00042
00043 const QSqlDatabase &DatabaseHandler::getActiveDatabase() const
00044 {
00045     return activeDatabase;
00046 }
00047
00048 const QSqlError &DatabaseHandler::getSqlError() const
00049 {
00050     return lastSqlError;
00051 }
00052
00053 void DatabaseHandler::setDatabasePath(const QString &newDatabasePath)
00054 {
00055     databasePath = newDatabasePath;
00056 }
00057
00058 void DatabaseHandler::setLastSqlError(const QSqlError &newLastSqlError)
00059 {
00060     lastSqlError = newLastSqlError;
00061 }

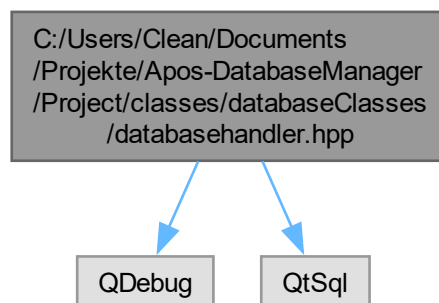
```

8.11 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/↵ Project/classes/databaseClasses/databasehandler.hpp File Reference

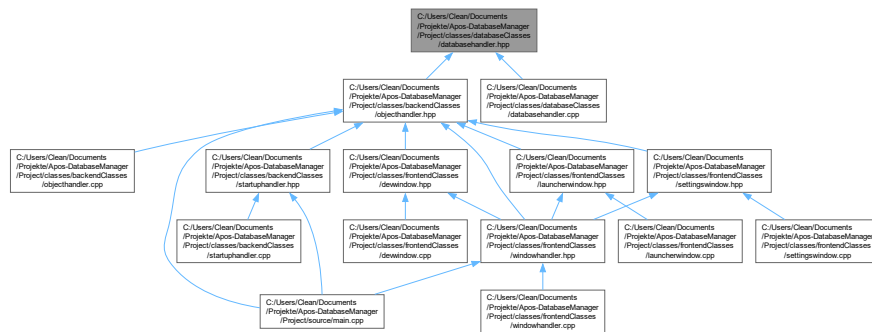
```
#include <QDebug>
```

```
#include <QtSql>
```

Include dependency graph for databasehandler.hpp:



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



Classes

- class [DatabaseHandler](#)

8.12 databasehandler.hpp

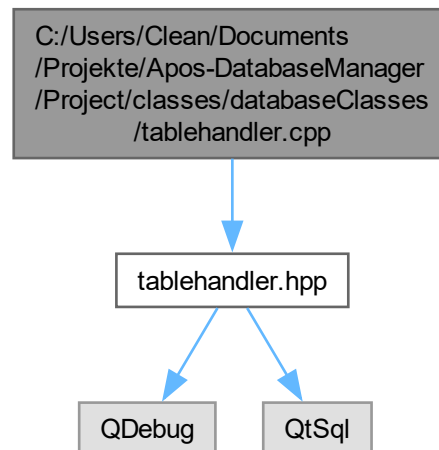
[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #include <QDebug>
00004 #include <QtSql>
00005
00006
00007 class DatabaseHandler
00008 {
00009 public:
00010
00011     DatabaseHandler();
00012
00013
00014     bool initDatabase();
00015
00016
00017     void closeDatabase();
00018
00019
00020     bool executeCommand(QString command);
00021
00022     const QSqlDatabase &getActiveDatabase() const;
00023
00024
00025     const QSqlError &getSqlError() const;
00026
00027
00028     void setDatabasePath(const QString &newDatabasePath);
00029
00030
00031
00032     void setLastSqlError(const QSqlError &newLastSqlError);
00033
00034 private:
00035
00036     QSqlError lastSqlError;
00037
00038     QSqlDatabase activeDatabase;
00039
00040     QString databasePath =
00041         R"(C:\Users\Clean\Documents\Projekte\Apos-DatabaseManager\Project\resources\defaultDatabase\userDatabase.db)";
00042
00043 };
```

8.13 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/↵ Project/classes/databaseClasses/tablehandler.cpp File Reference

#include "tablehandler.hpp"

Include dependency graph for tablehandler.cpp:



8.14 tablehandler.cpp

[Go to the documentation of this file.](#)

```

00001 #include "tablehandler.hpp"
00002
00003 TableHandler::TableHandler()
00004 {
00005
00006 }
00007
00008 TableHandler::TableHandler(const QSqlDatabase& newActiveDatabase, const QString &tableName)
00009 {
00010     tblActiveDatabase = newActiveDatabase;
00011     activeTableName = tableName;
00012     tableModel = new QSqlTableModel(nullptr, tblActiveDatabase);
00013     tableModel->setTable(tableName);
00014     tableModel->select();
00015 }
00016
00017 void TableHandler::generateTableModel()
00018 {
00019     tableModel = new QSqlTableModel(nullptr, tblActiveDatabase);
00020     tableModel->setTable(activeTableName);
00021     tableModel->select();
00022 }
00023
00024 void TableHandler::generateTableModel(QSqlDatabase& activeDatabase, const QString &tableName)
00025 {
00026     tblActiveDatabase = activeDatabase;
00027     activeTableName = tableName;
00028     tableModel = new QSqlTableModel(nullptr, activeDatabase);
00029     tableModel->setTable(tableName);
00030     tableModel->select();
00031 }
00032
00033 bool TableHandler::insertIntoTable(const QString &tableName, const QString &value1, const QString
&value2,
00034                                   const QString &value3, const QString &value4, const QString
&value5)
00035 {
00036     QSqlQuery query(tblActiveDatabase);
  
```



```

00037     query.prepare(QString("INSERT INTO %1 VALUES (:value1, :value2, :value3, :value4,
:value5)").arg(tableName));
00038     query.bindValue(":value1", value1);
00039     query.bindValue(":value2", value2);
00040     query.bindValue(":value3", value3);
00041     query.bindValue(":value4", value4);
00042     query.bindValue(":value5", value5);
00043     if(!query.exec()){
00044         lastTableError = query.lastError();
00045         return false;
00046     }
00047     return true;
00048 }
00049
00050 const QString &TableHandler::getActiveTableName() const
00051 {
00052     return activeTableName;
00053 }
00054
00055 void TableHandler::setActiveTableName(const QString &newActiveTableName)
00056 {
00057     activeTableName = newActiveTableName;
00058 }
00059
00060 const QSqlDatabase &TableHandler::getTblActiveDatabase() const
00061 {
00062     return tblActiveDatabase;
00063 }
00064
00065 const QSqlError &TableHandler::getLastTableError() const
00066 {
00067     return lastTableError;
00068 }
00069
00070 QSqlTableModel *TableHandler::getTableModel() const
00071 {
00072     return tableModel;
00073 }
00074

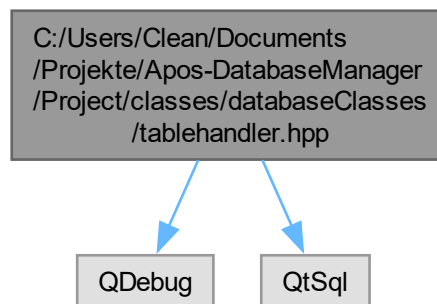
```

8.15 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/databaseClasses/tablehandler.hpp File Reference

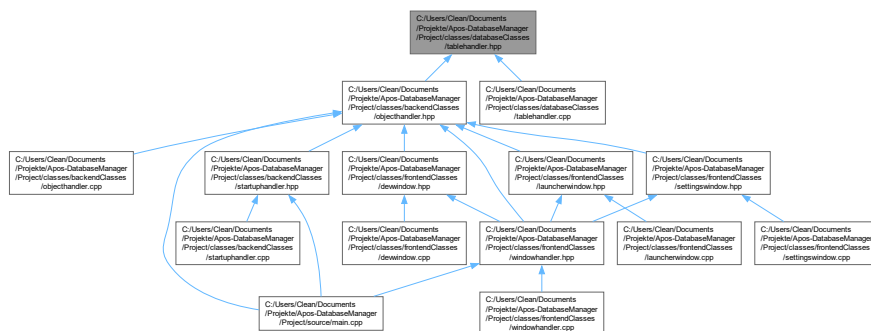
```
#include <QDebug>
```

```
#include <QtSql>
```

Include dependency graph for tablehandler.hpp:



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



Classes

- class [TableHandler](#)

8.16 tablehandler.hpp

[Go to the documentation of this file.](#)

```

00001 #pragma once
00002
00003 #include <QDebug>
00004 #include <QtSql>
00005
00006
00007 class TableHandler
00008 {
00009 public:
00010
00011     TableHandler();
00012
00013     TableHandler(const QSqlDatabase& newActiveDatabase, const QString& tableName);
00014
00015     void generateTableModel();
00016
00017     void generateTableModel(QSqlDatabase& activeDatabase, const QString &tableName);
00018
00019     QSqlTableModel *getTableModel() const;
00020
00021     const QSqlError &getTableSQLError() const;
00022
00023     void setTableName(const QString &newTableName);
00024
00025     void setActiveDatabase(QSqlDatabase newActiveDatabase);
00026
00027     bool insertIntoTable(const QString &tableName, const QString &value1, const QString &value2,
00028         const QString &value3, const QString &value4, const QString &value5);
00029
00030     const QString &getActiveTableName() const;
00031     void setActiveTableName(const QString &newActiveTableName);
00032
00033     const QSqlDatabase &getTblActiveDatabase() const;
00034     const QSqlError &getLastTableError() const;
00035 private:
00036

```

```

00053     QString activeTableName = "userTable";
00054
00055     QSqlTableModel *tableModel;
00056
00057     QSqlDatabase tblActiveDatabase;
00058
00059     QSqlError lastTableError;
00060 };
00061
00062

```

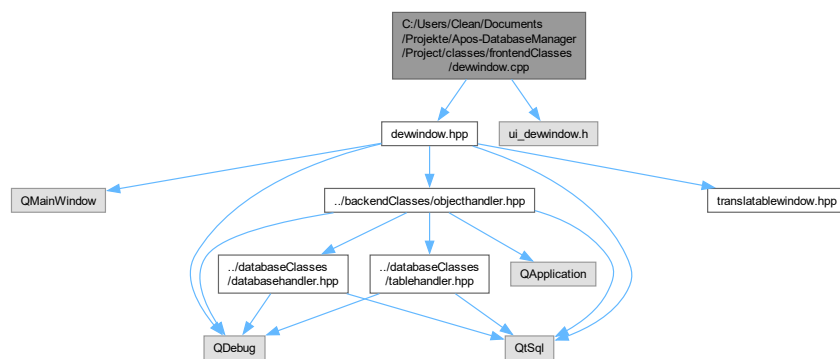
8.17 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/devwindow.cpp File Reference

```

#include "devwindow.hpp"
#include "ui_devwindow.h"

```

Include dependency graph for devwindow.cpp:



8.18 devwindow.cpp

[Go to the documentation of this file.](#)

```

00001 #include "devwindow.hpp"
00002 #include "ui_devwindow.h"
00003
00004
00005
00006 DevWindow::DevWindow(QWidget *parent, ObjectHandler *newObjectHandler)
00007     : QMainWindow(parent)
00008     , ui(new Ui::DevWindow)
00009 {
00010     ui->setupUi(this);
00011     objectHandler = newObjectHandler;
00012 }
00013
00014 DevWindow::~DevWindow()
00015 {
00016     delete ui;
00017 }
00018
00019 void DevWindow::logEvent(QString type, QString message)
00020 {
00021     ui->outLog->append("Log | " + type + ": " + message);
00022     qDebug() << "Logged: " + type + " - " + message;
00023 }
00024
00025 void DevWindow::logEvent(QString message, QSqlError error)
00026 {
00027     ui->outLog->append("Log | " + message + "-" + error.text());
00028     qDebug() << "Logged: " + message + " - " + error.text();
00029 }
00030
00031 void DevWindow::logEvent(QString message)
00032 {
00033     ui->outLog->append("Log | " + message);
00034     qDebug() << "Logged: " + message;

```

```

00035 }
00036
00037 void DevWindow::enableButtons(bool databaseLoaded)
00038 {
00039     ui->btnAdd->setEnabled(databaseLoaded);
00040     ui->btnCloseDB->setEnabled(databaseLoaded);
00041     ui->btnUpdate->setEnabled(databaseLoaded);
00042     ui->btnExecute->setEnabled(databaseLoaded);
00043     ui->btnSelectTable->setEnabled(databaseLoaded);
00044     ui->btnInitDB->setEnabled(!databaseLoaded);
00045     logEvent("action", "Buttons enabled/disabled");
00046 }
00047
00048 void DevWindow::setModelViews(QSqlTableModel &m)
00049 {
00050     ui->outTable->setModel(&m);
00051     ui->outColumn->setModel(&m);
00052     ui->outList->setModel(&m);
00053 }
00054
00055 void DevWindow::setModelViews()
00056 {
00057     ui->outTable->setModel(nullptr);
00058     ui->outColumn->setModel(nullptr);
00059     ui->outList->setModel(nullptr);
00060 }
00061
00062 void DevWindow::assignInputs()
00063 {
00064     input1 = ui->inInput1->text();
00065     input2 = ui->inInput2->text();
00066     input3 = ui->inInput3->text();
00067     input4 = ui->inInput4->text();
00068     input5 = ui->inInput5->text();
00069 }
00070
00071 void DevWindow::initDatabase(ObjectHandler *oH)
00072 {
00073     if(!oH->initDatabaseObject()){
00074         logEvent("Error initiating database", oH->getDbHandler()->getSqlError());
00075         return;
00076     }
00077     logEvent("action", "Database initiated");
00078     logEvent("action", QString(oH->getDbHandler()->getActiveDatabase().databaseName() + " opened");
00079     logEvent("status", oH->getDbHandler()->getActiveDatabase().isOpen() ? "Database open" : "Database
closed");
00080 }
00081
00082 void DevWindow::closeDatabase(DatabaseHandler *db)
00083 {
00084     db->closeDatabase();
00085     logEvent("action", "Database closed");
00086     logEvent("status", objectHandler->getDbHandler()->getActiveDatabase().isOpen() ? "Database open" :
"Database closed");
00087 }
00088
00089 bool DevWindow::checkCheckbox(int argCB)
00090 {
00091     if(argCB == 2){
00092         return true;
00093     }
00094     else if (argCB == 0){
00095         return false;
00096     }
00097     else {
00098         logEvent("warning", "Something went wrong!");
00099         return false;
00100     }
00101 }
00102
00103 void DevWindow::clearInputs(bool clearBool)
00104 {
00105     if(clearBool){
00106         ui->inInput1->clear();
00107         ui->inInput2->clear();
00108         ui->inInput3->clear();
00109         ui->inInput4->clear();
00110         ui->inInput5->clear();
00111         logEvent("action", "cleared inputs");
00112     }
00113 }
00114
00115 void DevWindow::clearCommandBox(bool clearBool)
00116 {
00117     if(clearBool){
00118         ui->inCommand->clear();
00119         logEvent("action", "tried to clear command-line");

```

```

00120     }
00121 }
00122
00123 void DevWindow::on_btnInitDB_clicked()
00124 {
00125     initDatabase(objectHandler);
00126     qDebug() << "Database initialized";
00127     objectHandler->initTableObject(objectHandler->getActiveDatabase(), "userTable");
00128     qDebug() << "TableHandler initialized";
00129     setModelViews(*objectHandler->getTableHandler()->getTableModel());
00130     qDebug() << "ModelViews set";
00131     enableButtons(true);
00132 }
00133
00134 void DevWindow::on_btnCloseDB_clicked()
00135 {
00136     setModelViews();
00137     closeDatabase(objectHandler->getDbHandler());
00138     enableButtons(false);
00139 }
00140
00141 void DevWindow::on_btnExecute_clicked()
00142 {
00143     if(!objectHandler->getDbHandler()->executeCommand(ui->inCommand->toPlainText())){
00144         logEvent("Error executing command", objectHandler->getDbHandler()->getSqlError());
00145         clearCommandBox(clearCommand);
00146         return;
00147     }
00148     clearCommandBox(clearCommand);
00149     logEvent("action", "Command executed");
00150 }
00151
00152 void DevWindow::on_btnSelectTable_clicked()
00153 {
00154     objectHandler->getTableHandler()->generateTableModel();
00155     logEvent("action", "Table selected");
00156     setModelViews(*objectHandler->getTableHandler()->getTableModel());
00157 }
00158
00159 void DevWindow::on_btnAdd_clicked()
00160 {
00161     assignInputs();
00162     if(!objectHandler->getTableHandler()->insertIntoTable(objectHandler->getActiveTableName(), input1,
00163 input2, input3, input4, input5)){
00164         logEvent("Insert Error", objectHandler->getTableSqlError());
00165         clearInputs(clearInput);
00166         return;
00167     }
00168     clearInputs(clearInput);
00169     logEvent("action", "Values inserted");
00170 }
00171 void DevWindow::on_btnUpdate_clicked()
00172 {
00173     // Generating table model before updating the views
00174     objectHandler->getTableHandler()->generateTableModel();
00175     setModelViews(*objectHandler->getTableHandler()->getTableModel());
00176     logEvent("action", "Table view updated");
00177 }
00178
00179 void DevWindow::on_clearCommandAfterExecute_stateChanged(int arg1)
00180 {
00181     clearCommand = checkCheckbox(arg1);
00182     logEvent("status", "Command will clear after execute: " + QString(clearCommand ? "true" :
00183 "false"));
00184 }
00185 void DevWindow::on_clearInputsAfterInsert_stateChanged(int arg1)
00186 {
00187     clearInput = checkCheckbox(arg1);
00188     logEvent("status", "Inputs will be cleared after execution: " + QString(clearInput ? "true" :
00189 "false"));
00190 }
00191 void DevWindow::on_inReturnToLauncher_clicked()
00192 {
00193     emit returnToLauncher();
00194 }
00195
00196 void DevWindow::retranslateUi()
00197 {
00198     ui->retranslateUi(this);
00199 }
00200
00201 void DevWindow::on_inSettings_clicked()
00202 {
00203     emit openSettings();

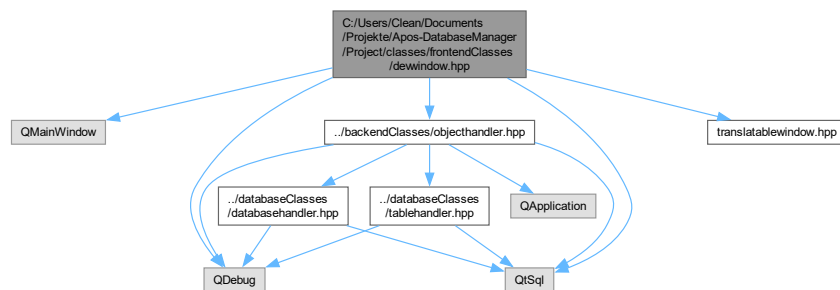
```

```
00204 }
00205
```

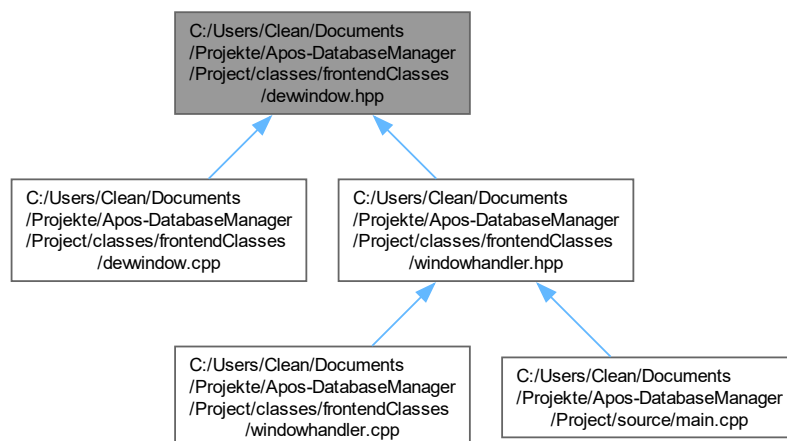
8.19 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/↵ Project/classes/frontendClasses/devwindow.hpp File Reference

```
#include <QMainWindow>
#include <QDebug>
#include <QtSql>
#include "../backendClasses/objecthandler.hpp"
#include "translatablewindow.hpp"
```

Include dependency graph for devwindow.hpp:



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



Classes

- class [DevWindow](#)

Namespaces

- namespace [Ui](#)

8.20 devwindow.hpp

[Go to the documentation of this file.](#)

```

00001 #pragma once
00002
00003 #include <QMainWindow>
00004 #include <QDebug>
00005 #include <QtSql>
00006
00007 #include "../backendClasses/objecthandler.hpp"
00008 #include "translatablewindow.hpp"
00009
00010 QT_BEGIN_NAMESPACE
00011 namespace Ui { class DevWindow; }
00012 QT_END_NAMESPACE
00013
00014 class DevWindow : public QMainWindow, public TranslatableWindow
00015 {
00016     Q_OBJECT
00017
00018 public:
00019     DevWindow(QWidget *parent = nullptr, ObjectHandler *objectHandler = nullptr);
00020
00021     ~DevWindow();
00022
00023     void logEvent(QString type, QString message);
00024
00025     void logEvent(QString message, QSqlError error);
00026
00027     void logEvent(QString message);
00028
00029     void retranslateUi();
00030
00031 signals:
00032     void returnToLauncher();
00033     void openSettings();
00034
00035 private slots:
00036     void on_btnInitDB_clicked();
00037
00038     void on_btnCloseDB_clicked();
00039
00040     void on_btnExecute_clicked();
00041
00042     void on_btnSelectTable_clicked();
00043
00044     void on_btnAdd_clicked();
00045
00046     void on_btnUpdate_clicked();
00047
00048     void on_clearCommandAfterExecute_stateChanged(int arg1);
00049
00050     void on_clearInputsAfterInsert_stateChanged(int arg1);
00051
00052     void on_inReturnToLauncher_clicked();
00053
00054     void on_inSettings_clicked();
00055
00056 private:
00057     void enableButtons(bool databaseLoaded);
00058
00059     void setModelViews(QSqlTableModel &m);
00060
00061     void setModelViews();
00062
00063     void assignInputs();
00064
00065     bool checkCheckbox(int argCB);

```

```

00084
00085     void initDatabase(ObjectHandler* oH);
00086
00087     void closeDatabase(DatabaseHandler* db);
00088
00089     void clearInputs(bool clearBool);
00090
00091
00092     void clearCommandBox(bool clearBool);
00093
00094     Ui::DevWindow *ui;
00095     ObjectHandler *objectHandler;
00096
00097     QString input1, input2, input3, input4, input5;
00098     bool clearCommand = false, clearInput = false;
00099 };
00100
00101

```

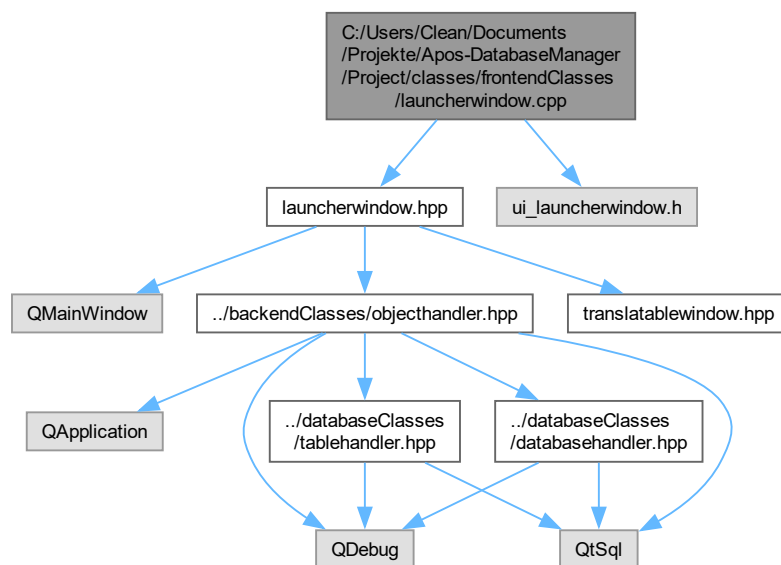
8.21 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/launcherwindow.cpp File Reference

```

#include "launcherwindow.hpp"
#include "ui_launcherwindow.h"

```

Include dependency graph for launcherwindow.cpp:



8.22 launcherwindow.cpp

[Go to the documentation of this file.](#)

```

00001 #include "launcherwindow.hpp"
00002 #include "ui_launcherwindow.h"
00003
00004 LauncherWindow::LauncherWindow(QWidget *parent, ObjectHandler* newObjectHandler) :
00005     QMainWindow(parent),
00006     ui(new Ui::LauncherWindow)
00007 {
00008     ui->setupUi(this);
00009     objectHandler = newObjectHandler;
00010 }
00011

```



```

00012
00013 LauncherWindow::~LauncherWindow()
00014 {
00015     delete ui;
00016 }
00017
00018
00019 void LauncherWindow::on_inShowDev_clicked()
00020 {
00021     emit openDevWindow();
00022 }
00023
00024
00025
00026
00027 void LauncherWindow::on_pushButton_clicked()
00028 {
00029     emit openSettings();
00030 }
00031
00032 void LauncherWindow::retranslateUi()
00033 {
00034
00035 }
00036

```

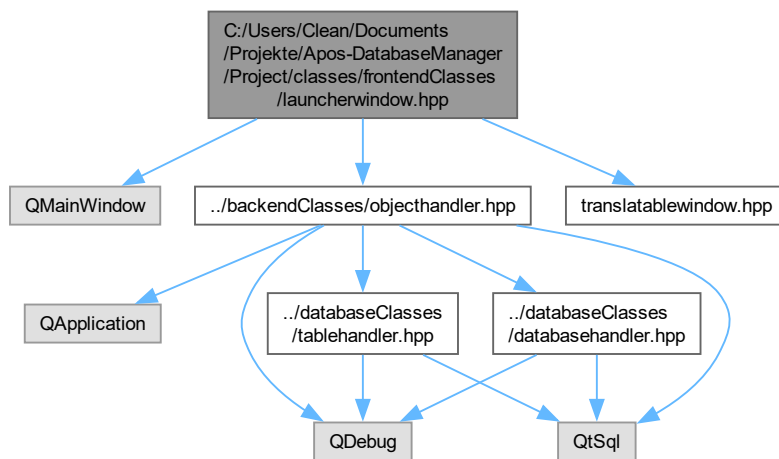
8.23 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/launcherwindow.hpp File Reference

```

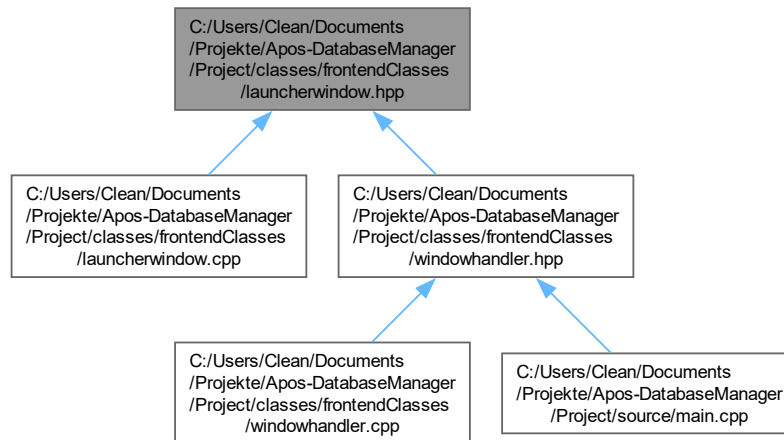
#include <QMainWindow>
#include "../backendClasses/objecthandler.hpp"
#include "translatablewindow.hpp"

```

Include dependency graph for launcherwindow.hpp:



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



Classes

- class [LauncherWindow](#)

Namespaces

- namespace [Ui](#)

8.24 launcherwindow.hpp

[Go to the documentation of this file.](#)

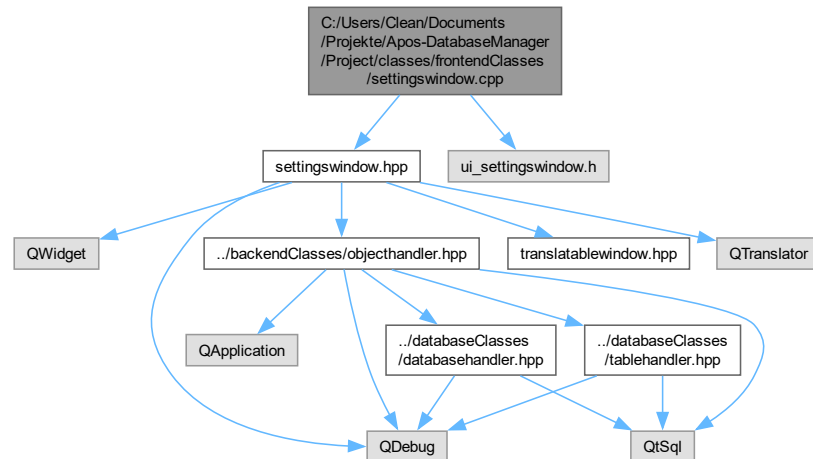
```

00001 #pragma once
00002
00003 #include <QMainWindow>
00004
00005 #include "../backendClasses/objecthandler.hpp"
00006 #include "translatablewindow.hpp"
00007
00008 namespace Ui {
00009     class LauncherWindow;
00010 }
00011
00012 class LauncherWindow : public QMainWindow, public TranslatableWindow
00013 {
00014     Q_OBJECT
00015
00016 public:
00017     explicit LauncherWindow(QWidget *parent = nullptr, ObjectHandler* newObjectHandler = nullptr);
00018     ~LauncherWindow();
00019
00020     Ui::LauncherWindow *ui;
00021     void retranslateUi();
00022 signals:
00023     void openDevWindow();
00024     void openSettings();
00025 private slots:
00026     void on_inShowDev_clicked();
00027     void on_pushButton_clicked();
00028
00029 private:
00030     ObjectHandler* objectHandler;
00031 };
00032
00033

```

8.25 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/settingswindow.cpp File Reference

```
#include "settingswindow.hpp"
#include "ui_settingswindow.h"
Include dependency graph for settingswindow.cpp:
```



8.26 settingswindow.cpp

[Go to the documentation of this file.](#)

```
00001 #include "settingswindow.hpp"
00002 #include "ui_settingswindow.h"
00003
00004 SettingsWindow::SettingsWindow(QWidget *parent, ObjectHandler *newObjectHandler) :
00005     QWidget(parent),
00006     ui(new Ui::SettingsWindow)
00007 {
00008     ui->setupUi(this);
00009     objectHandler = newObjectHandler;
00010     translator = new QTranslator;
00011 }
00012
00013 SettingsWindow::~SettingsWindow()
00014 {
00015     delete ui;
00016 }
00017
00018 void SettingsWindow::retranslateUi()
00019 {
00020     ui->retranslateUi(this);
00021 }
00022
00023 void SettingsWindow::on_inClose_clicked()
00024 {
00025     this->hide();
00026 }
00027
00028
00029 void SettingsWindow::on_inApply_clicked()
00030 {
00031     if(languageChanged){
00032         qDebug() << "New Language will be applied";
00033         languageIndex = tempLanguageIndex;
00034         installTranslator();
00035     }
00036     emit appliedSettings();
00037 }
00038
00039
00040
00041 void SettingsWindow::on_inLanguage_currentIndexChanged(int index)
```

```

00042 {
00043     tempLanguageIndex = index;
00044     qDebug() << "checkboxIndex: " << index << "tempIndex: " << tempLanguageIndex << "index: " <<
    languageIndex;
00045     if(tempLanguageIndex == languageIndex){
00046         qDebug() << "Language not changed";
00047         languageChanged = false;
00048         return;
00049     }
00050     qDebug() << "Language changed";
00051     languageChanged = true;
00052 }
00053
00054 void SettingsWindow::installTranslator()
00055 {
00056     qDebug() << languageIndex;
00057     objectHandler->application->removeTranslator(translator);
00058     delete translator;
00059     translator = new QTranslator;
00060     switch (languageIndex){
00061     case 0:
00062         translator->load(":/i18n/Apos-DatabaseManager_en_GB");
00063         qDebug() << "tried to load english";
00064         objectHandler->application->installTranslator(translator);
00065         break;
00066     case 1:
00067         translator->load(":/i18n/Apos-DatabaseManager_de_DE");
00068         qDebug() << "tried to load german";
00069         objectHandler->application->installTranslator(translator);
00070         break;
00071     }
00072 }
00073

```

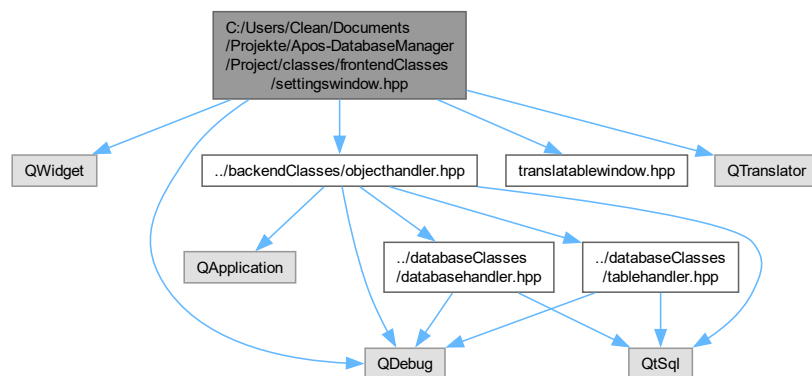
8.27 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/↵ Project/classes/frontendClasses/settingswindow.hpp File Reference

```

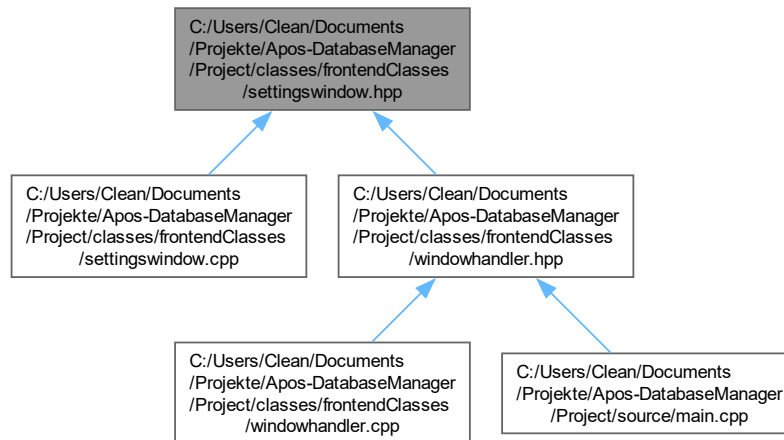
#include <QWidget>
#include <QDebug>
#include "../backendClasses/objecthandler.hpp"
#include "translatablewindow.hpp"
#include <QTranslator>

```

Include dependency graph for settingswindow.hpp:



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



Classes

- class [SettingsWindow](#)

Namespaces

- namespace [Ui](#)

8.28 settingswindow.hpp

[Go to the documentation of this file.](#)

```

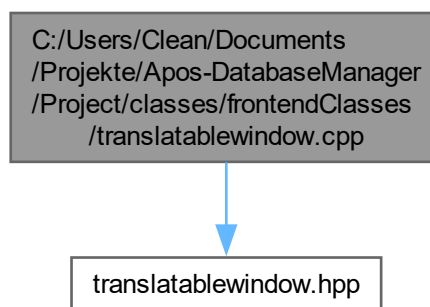
00001 #pragma once
00002
00003 #include <QWidget>
00004 #include <QDebug>
00005 #include "../backendClasses/objecthandler.hpp"
00006 #include "translatablewindow.hpp"
00007 #include <QTranslator>
00008
00009
00010 namespace Ui {
00011 class SettingsWindow;
00012 }
00013
00014 class SettingsWindow : public QWidget, public TranslatableWindow
00015 {
00016     Q_OBJECT
00017
00018 public:
00019     explicit SettingsWindow(QWidget *parent = nullptr, ObjectHandler *objectHandler = nullptr);
00020     ~SettingsWindow();
00021     void retranslateUi();
00022 signals:
00023     void appliedSettings();
00024
00025 private slots:
00026     void on_inClose_clicked();
00027
00028     void on_inApply_clicked();
00029
00030     void on_inLanguage_currentIndexChanged(int index);
00031
00032 private:
00033     void installTranslator();
00034
00035     Ui::SettingsWindow *ui;
00036     QString language;
00037
  
```

```
00038     ObjectHandler* objectHandler;  
00039     QTranslator* translator;  
00040     int languageIndex, tempLanguageIndex;  
00041     bool languageChanged;  
00042 };  
00043
```

8.29 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/↔ Project/classes/frontendClasses/translatablewindow.cpp File Reference

```
#include "translatablewindow.hpp"
```

Include dependency graph for translatablewindow.cpp:



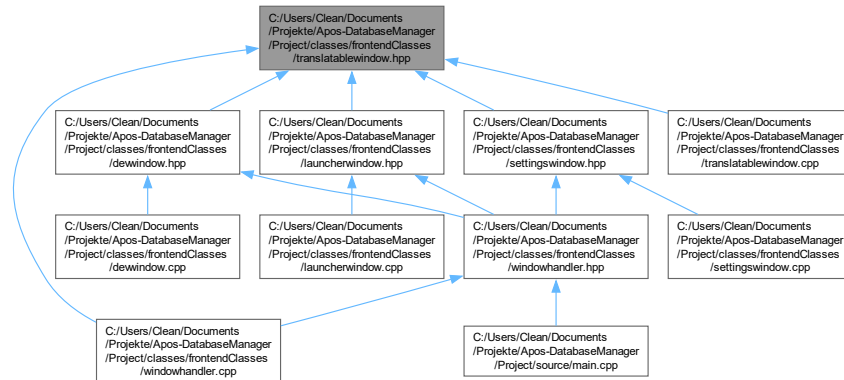
8.30 translatablewindow.cpp

[Go to the documentation of this file.](#)

```
00001 #include "translatablewindow.hpp"  
00002  
00003 TranslatableWindow::TranslatableWindow()  
00004 {  
00005  
00006 }
```

8.31 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/translatewindow.hpp File Reference

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



Classes

- class [TranslatableWindow](#)

8.32 translatewindow.hpp

[Go to the documentation of this file.](#)

```

00001 #pragma once
00002
00003
00004 class TranslatableWindow
00005 {
00006 public:
00007     TranslatableWindow();
00008     virtual void retranslateUi() = 0;
00009 };
00010

```

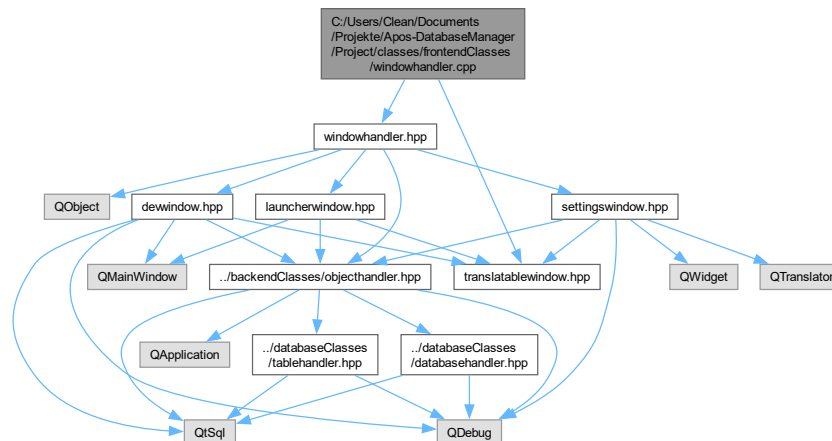
8.33 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/windowhandler.cpp File Reference

```

#include "windowhandler.hpp"
#include "translatewindow.hpp"

```

Include dependency graph for windowhandler.cpp:



8.34 windowhandler.cpp

[Go to the documentation of this file.](#)

```

00001 #include "windowhandler.hpp"
00002 #include "translatablewindow.hpp"
00003
00004
00005 WindowHandler::WindowHandler(ObjectHandler *newObjectHandler)
00006 {
00007
00008     objectHandler = newObjectHandler;
00009     launcherWindow = new LauncherWindow(nullptr, objectHandler);
00010     settingsWindow = new SettingsWindow();
00011     QObject::connect(launcherWindow, &LauncherWindow::openDevWindow, this,
00012 &WindowHandler::showDevWindow, Qt::DirectConnection);
00012     QObject::connect(launcherWindow, &LauncherWindow::openSettings, this,
00013 &WindowHandler::showSettingsWindow);
00013     QObject::connect(settingsWindow, &SettingsWindow::appliedSettings, this,
00014 &WindowHandler::applySettings);
00014
00015 }
00016
00017 void WindowHandler::showLaunchWindow()
00018 {
00019     if (devWindow != nullptr) {
00020         if (!devWindow->isHidden()) {
00021             devWindow->hide();
00022         }
00023     }
00024     launcherWindow->show();
00025 }
00026
00027 void WindowHandler::showDevWindow()
00028 {
00029     launcherWindow->hide();
00030     if (devWindow == nullptr) {
00031         devWindow = new DevWindow(nullptr, objectHandler);
00032         QObject::connect(devWindow, &DevWindow::returnToLauncher, this,
00033 &WindowHandler::showLaunchWindow, Qt::DirectConnection);
00033         QObject::connect(devWindow, &DevWindow::openSettings, this,
00034 &WindowHandler::showSettingsWindow, Qt::DirectConnection);
00034     }
00035     devWindow->show();
00036 }
00037
00038 void WindowHandler::showSettingsWindow()
00039 {
00040     settingsWindow->show();
00041 }
00042
00043 void WindowHandler::applySettings()
00044 {
00045     changeLanguages();
00046 }
  
```



```

00047 }
00048
00049 void WindowHandler::changeLanguages()
00050 {
00051     QList<QWidget*> openWindows = objectHandler->application->topLevelWidgets();
00052     for(QWidget *widget : qAsConst(openWindows)){
00053         TranslatableWindow *tw = dynamic_cast<TranslatableWindow*>(widget);
00054         if(tw){
00055             qDebug() << "Dynamic cast pointer adress " << tw;
00056             tw->retranslateUi();
00057         }
00058     }
00059     qDebug() << openWindows;
00060 }

```

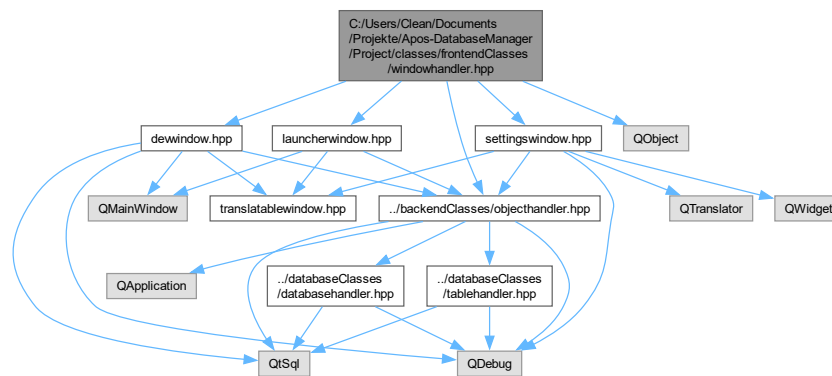
8.35 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/Project/classes/frontendClasses/windowhandler.hpp File Reference

```

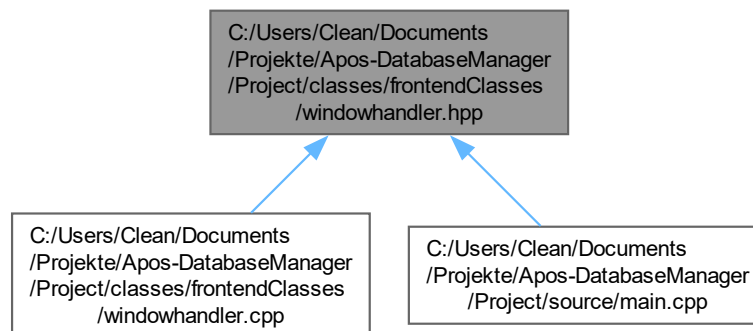
#include "launcherwindow.hpp"
#include "devwindow.hpp"
#include "../backendClasses/objecthandler.hpp"
#include "settingswindow.hpp"
#include <QObject>

```

Include dependency graph for windowhandler.hpp:



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



Classes

- class [WindowHandler](#)

8.36 windowhandler.hpp

[Go to the documentation of this file.](#)

```

00001 #pragma once
00002
00003 #include "launcherwindow.hpp"
00004 #include "devwindow.hpp"
00005 #include "../backendClasses/objecthandler.hpp"
00006 #include "settingswindow.hpp"
00007 #include <QObject>
00008
00009 class WindowHandler : public QObject
00010 {
00011 public:
00012     WindowHandler(ObjectHandler* newObjectHandler);
00013
00014
00015 public slots:
00016     void showLaunchWindow();
00017
00018 private slots:
00019     void showDevWindow();
00020     void showSettingsWindow();
00021     void applySettings();
00022
00023 private:
00024
00025     void changeLanguages();
00026
00027
00028     LauncherWindow *launcherWindow = nullptr;
00029     DevWindow *devWindow = nullptr;
00030     SettingsWindow *settingsWindow = nullptr;
00031     ObjectHandler *objectHandler;
00032
00033 };
00034

```

**8.37 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/↵
Project/README.md File Reference****8.38 C:/Users/Clean/Documents/Projekte/Apos-DatabaseManager/↵
Project/source/main.cpp File Reference**

```

#include "../classes/backendClasses/startuphandler.hpp"
#include "../classes/backendClasses/objecthandler.hpp"
#include "../classes/frontendClasses/windowhandler.hpp"
#include <QApplication>
#include <QDebug>
#include <QtSql>

```



```
00014
00015     StartupHandler startupHandler(a);
00016     qDebug() << "StartupHandler Object initialized";
00017
00018     ObjectHandler *objectHandler = startupHandler.startUp();
00019     qDebug() << "ObjectHandler Object initialized";
00020     startupHandler.~StartupHandler();
00021
00022     WindowHandler windowHandler(objectHandler);
00023     windowHandler.showLaunchWindow();
00024
00025     qDebug() << "After DevWindow Show";
00026
00027     return a->exec();
00028 }
```


- clearCommand, 28
- clearCommandBox, 18
- clearInput, 28
- clearInputs, 19
- closeDatabase, 19
- DevWindow, 17
- enableButtons, 20
- initDatabase, 21
- input1, 28
- input2, 28
- input3, 28
- input4, 28
- input5, 29
- logEvent, 21, 22
- objectHandler, 29
- on_btnAdd_clicked, 22
- on_btnCloseDB_clicked, 23
- on_btnExecute_clicked, 23
- on_btnInitDB_clicked, 24
- on_btnSelectTable_clicked, 24
- on_btnUpdate_clicked, 25
- on_clearCommandAfterExecute_stateChanged, 25
- on_clearInputsAfterInsert_stateChanged, 26
- on_inReturnToLauncher_clicked, 26
- on_inSettings_clicked, 26
- openSettings, 27
- retranslateUi, 27
- returnToLauncher, 27
- setModelViews, 27, 28
- ui, 29
- devWindow
 - WindowHandler, 53
- enableButtons
 - DevWindow, 20
- executeCommand
 - DatabaseHandler, 14
- generateTableModel
 - TableHandler, 45
- getActiveDatabase
 - DatabaseHandler, 14
 - ObjectHandler, 33
- getActiveTableName
 - ObjectHandler, 34
 - TableHandler, 46
- getDbHandler
 - ObjectHandler, 34
- getLastTableError
 - TableHandler, 46
- getSqlError
 - DatabaseHandler, 14
- getTableHandler
 - ObjectHandler, 35
- getTableModel
 - TableHandler, 46
- getTableSqlError
 - ObjectHandler, 35
- getTblActiveDatabase
 - TableHandler, 47
- initDatabase
 - DatabaseHandler, 15
 - DevWindow, 21
- initDatabaseObject
 - ObjectHandler, 36
- initObjectHandler
 - StartupHandler, 43
- initTableObject
 - ObjectHandler, 36, 37
- initTranslator
 - StartupHandler, 43
- input1
 - DevWindow, 28
- input2
 - DevWindow, 28
- input3
 - DevWindow, 28
- input4
 - DevWindow, 28
- input5
 - DevWindow, 29
- insertIntoTable
 - TableHandler, 47
- installTranslator
 - SettingsWindow, 40
- language
 - SettingsWindow, 41
- languageChanged
 - SettingsWindow, 41
- languageIndex
 - SettingsWindow, 41
- lastSqlError
 - DatabaseHandler, 15
- lastTableError
 - TableHandler, 48
- LauncherWindow, 29
 - ~LauncherWindow, 30
 - LauncherWindow, 30
 - objectHandler, 32
 - on_inShowDev_clicked, 31
 - on_pushButton_clicked, 31
 - openDevWindow, 31
 - openSettings, 31
 - retranslateUi, 32
 - ui, 32
- launcherWindow
 - WindowHandler, 53
- logEvent
 - DevWindow, 21, 22
- main
 - main.cpp, 83
- main.cpp

- main, [83](#)
- ObjectHandler, [32](#)
 - application, [37](#)
 - dbHandler, [37](#)
 - getActiveDatabase, [33](#)
 - getActiveTableName, [34](#)
 - getDbHandler, [34](#)
 - getTableHandler, [35](#)
 - getTableSqlError, [35](#)
 - initDatabaseObject, [36](#)
 - initTableObject, [36, 37](#)
 - ObjectHandler, [33](#)
 - setActiveTableName, [37](#)
 - tableHandler, [38](#)
- objectHandler
 - DevWindow, [29](#)
 - LauncherWindow, [32](#)
 - SettingsWindow, [41](#)
 - StartupHandler, [44](#)
 - WindowHandler, [53](#)
- on_btnAdd_clicked
 - DevWindow, [22](#)
- on_btnCloseDB_clicked
 - DevWindow, [23](#)
- on_btnExecute_clicked
 - DevWindow, [23](#)
- on_btnInitDB_clicked
 - DevWindow, [24](#)
- on_btnSelectTable_clicked
 - DevWindow, [24](#)
- on_btnUpdate_clicked
 - DevWindow, [25](#)
- on_clearCommandAfterExecute_stateChanged
 - DevWindow, [25](#)
- on_clearInputsAfterInsert_stateChanged
 - DevWindow, [26](#)
- on_inApply_clicked
 - SettingsWindow, [40](#)
- on_inClose_clicked
 - SettingsWindow, [40](#)
- on_inLanguage_currentIndexChanged
 - SettingsWindow, [40](#)
- on_inReturnToLauncher_clicked
 - DevWindow, [26](#)
- on_inSettings_clicked
 - DevWindow, [26](#)
- on_inShowDev_clicked
 - LauncherWindow, [31](#)
- on_pushButton_clicked
 - LauncherWindow, [31](#)
- openDevWindow
 - LauncherWindow, [31](#)
- openSettings
 - DevWindow, [27](#)
 - LauncherWindow, [31](#)
- ptrApplication
 - StartupHandler, [44](#)
- retranslateUi
 - DevWindow, [27](#)
 - LauncherWindow, [32](#)
 - SettingsWindow, [41](#)
 - TranslatableWindow, [49](#)
- returnToLauncher
 - DevWindow, [27](#)
- setActiveDatabase
 - TableHandler, [47](#)
- setActiveTableName
 - ObjectHandler, [37](#)
 - TableHandler, [47](#)
- setDatabasePath
 - DatabaseHandler, [15](#)
- setLastSqlError
 - DatabaseHandler, [15](#)
- setModelViews
 - DevWindow, [27, 28](#)
- setTableName
 - TableHandler, [47](#)
- SettingsWindow, [38](#)
 - ~SettingsWindow, [39](#)
 - appliedSettings, [39](#)
 - installTranslator, [40](#)
 - language, [41](#)
 - languageChanged, [41](#)
 - languageIndex, [41](#)
 - objectHandler, [41](#)
 - on_inApply_clicked, [40](#)
 - on_inClose_clicked, [40](#)
 - on_inLanguage_currentIndexChanged, [40](#)
 - retranslateUi, [41](#)
 - SettingsWindow, [39](#)
 - tempLanguageIndex, [41](#)
 - translator, [41](#)
 - ui, [41](#)
- settingsWindow
 - WindowHandler, [53](#)
- showDevWindow
 - WindowHandler, [51](#)
- showLaunchWindow
 - WindowHandler, [52](#)
- showSettingsWindow
 - WindowHandler, [52](#)
- startUp
 - StartupHandler, [43](#)
- StartupHandler, [41](#)
 - ~StartupHandler, [42](#)
 - initObjectHandler, [43](#)
 - initTranslator, [43](#)
 - objectHandler, [44](#)
 - ptrApplication, [44](#)
 - startUp, [43](#)
 - StartupHandler, [42](#)
- TableHandler, [44](#)
 - activeTableName, [48](#)
 - generateTableModel, [45](#)

- getActiveTableName, [46](#)
- getLastTableError, [46](#)
- getTableModel, [46](#)
- getTableSQLException, [47](#)
- getTblActiveDatabase, [47](#)
- insertIntoTable, [47](#)
- lastTableError, [48](#)
- setActiveDatabase, [47](#)
- setActiveTableName, [47](#)
- setTableName, [47](#)
- TableHandler, [45](#)
- tableModel, [48](#)
- tblActiveDatabase, [48](#)
- tableHandler
 - ObjectHandler, [38](#)
- tableModel
 - TableHandler, [48](#)
- tblActiveDatabase
 - TableHandler, [48](#)
- tempLanguageIndex
 - SettingsWindow, [41](#)
- TranslatableWindow, [48](#)
 - retranslateUi, [49](#)
 - TranslatableWindow, [48](#)
- translator
 - SettingsWindow, [41](#)
- Ui, [11](#)
- ui
 - DevWindow, [29](#)
 - LauncherWindow, [32](#)
 - SettingsWindow, [41](#)
- WindowHandler, [49](#)
 - applySettings, [50](#)
 - changeLanguages, [51](#)
 - devWindow, [53](#)
 - launcherWindow, [53](#)
 - objectHandler, [53](#)
 - settingsWindow, [53](#)
 - showDevWindow, [51](#)
 - showLaunchWindow, [52](#)
 - showSettingsWindow, [52](#)
 - WindowHandler, [50](#)