

Computer Science Advising Tool Database Handler Setup

Last Place Champions
Rico Adrian, Brian Hooper, Nick Rohde

Last Updated: 8th of March 2018

Contents

1	Overview	2
1.1	Requirements	2
2	Compiling	2
2.1	Compiling on Windows	2
2.2	Compiling on Linux	3
2.3	Next Steps	3
3	Creating the Service	3
3.1	Preparation	3
3.2	Putting Everything into Place	4
3.3	Starting DBH	4
4	The dbh.service File	5
4.1	Unit	5
4.2	Service	5
4.3	Install	5
5	The Configuration.ini File	5
5.1	MySQL Connection	6
5.2	MySQL Tables	6
5.3	Misc	7
6	MySQL Setup	7
7	Troubleshooting	8

1 Overview

This document explains how to compile, daemonize, and use the Database Handler service.

Chapter 2 explains the process of taking the source files provided by the Developers and compiling them into a runnable application.

Chapter 3 explains the process of creating the service, including the process of creating the dbh.service file, as well as where to place the service file, and other accompanying files in the system.

Chapter 4 explains the structure the dbh.ini file must follow in order to function properly, and what changes can be made by the administrator.

Chapter 5 explains the process of setting up the tables in the MySQL database, and details important structural requirements.

Chapter 6 explains some troubleshooting techniques that will help you solve issues that may occur during the setup process.

1.1 Requirements

In order to run the Database Handler, the server must be running on the Debian 9 Server operating system, and the dotnet core SDK version 2.0.x installed. Information on installing the dotnet core SDK on Debian can be found here: <https://docs.microsoft.com/en-us/dotnet/core/linux-prerequisites?tabs=netcore2x>

2 Compiling

This chapter goes over the process of compiling the provided source files into the assemblies which can be run by the dotnet runtime. It is assumed that the source files are in the same folder structure as they were supplied by the Developers, and all files are present.

2.1 Compiling on Windows

Compiling the Database Handler (DBH) on Windows is the simplest way to create the assemblies, and it is the recommended way of compiling.

1. Open a command prompt, and navigate to the directory containing the file CWU_Advising_Code.sln
2. Execute the command: **dotnet publish -c Release -r debian-x64**
The build output should look like this: ¡TODO INSERT PIC!

Now, the output directory should contain the directory "/netcoreapp2.0/publish". This publish directory contains all required files to run the DBH, and it should not be altered.

2.2 Compiling on Linux

Compiling the Database Handler (DBH) on linux must be done by manually compiling the Database_Object_Classes.csproj first, and after that the Database_Handler.csproj.

1. Open a terminal, and navigate to the directory containing the file Database_Object_Classes.csproj
2. Execute the command: **dotnet build -c Release**
The build output should look like this: ¡TODO INSERT PIC¿
3. Navigate to the directory containing the file Database_Handler.csproj
4. Execute the command: **dotnet publish -c Release**
The build output should look like this: ¡TODO INSERT PIC¿

Now, the output directory should contain a directory `"/output/netcoreapp2.0/debian-x64/publish"`. This publish directory contains all required files to run the DBH, and it should not be altered.

2.3 Next Steps

In the next steps, we will refer to the directory `"/output/netcoreapp2.0/debian-x64/publish"` as the publish directory. This directory will be used in Section 3.2 "Putting Everything into Place" to create the daemon that will run the DBH.

3 Creating the Service

3.1 Preparation

Now we will create the service to host the databases. First, we need to create the service file, and store it in the appropriate directory for the operating system to find it.

1. Find the file titled `"dbh.service"` which is in the same directory as this document, as well as the file `"Configuration.ini"`.
2. Place these two files in your home directory on the server using your method of choice.
3. Execute the command: **sudo mv dbh.service /etc/systemd/system/**
4. Execute the command: **sudo nano /etc/systemd/system/**
5. Ensure the following settings are in the file, and update them as necessary:
 - (a) `TODO INSERT PIC`
 - (b) `WorkingDirectory=/var/aspnetcore/publish`

- (c) `ExecStart=/usr/bin/dotnet /var/aspnetcore/publish/DatabaseHandler.dll`
 - (d) `Restart=always`
 - (e) `RestartSec=10`
 - (f) `Environment=ASPNETCORE_ENVIRONMENT=Development`
 - (g) `Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false`
 - (h) `WantedBy=multi-user.target`
 - (i) **Ensure that "User=" does not appear in the file, if it does remove it.**
6. Once you have verified the contents of the file, save the file by pressing CTRL+O, then press Enter, and finally exit the file by pressing CTRL+X

3.2 Putting Everything into Place

Next we need to place the publish directory (see 2.3 "Next Steps") in the correct place.

1. Go to your home directory (Execute the command: `"cd ~"`)
2. Execute the command: `sudo mkdir /var/aspnetcore/`
3. Execute the command: `sudo mv publish /var/aspnetcore/`
4. Execute the command: `sudo mv Configuration.ini /var/aspnetcore/publish/`

Now we have successfully created the service, and put all files into the necessary locations to start the database handler. Before we can start the service, we must first update an entry in the "Configuration.ini" file. To do this, see Section 5.1 "MySQLConnection" before going to section 3.3.

3.3 Starting DBH

1. Execute the command: `sudo systemctl daemon-reload`
2. Execute the command: `sudo systemctl enable dbh.service`
3. Execute the command: `sudo systemctl start dbh.service`
4. Execute the command: `sudo systemctl status dbh.service` You should now see output similar to this: `ïTODO INSERT PICï`
5. If the output reads "Active: active(running)" (usually in green colour), you're done. Otherwise, go to Section 5.1 for troubleshooting.

4 The dbh.service File

This file contains the definition of the Database Handler service, it must be present in the correct location (see 3.1 "Preparation" for more info) in order for the Database Handler to function properly. This section will explain the structure of this file, and how it can be changed, though we advise against changing the contents of this file.

4.1 Unit

The Unit section contains the general information about the service, by default, it should contain a short description. Feel free to modify this to a more fitting description for you. **Note: Only change the description value to the right of the equals sign, do not change the left side.**

4.2 Service

The Service section contains the actual definition of the service.

1. WorkingDirectory: This setting tells the service where to find all necessary files, if this setting is changed, all files from Section 3 must be moved to this location.
2. ExecStart: This setting tells the service how to run, and what file to run. The first setting, `"/usr/bin/dotnet"`, **must not be altered**, the second setting must be altered if the WorkingDirectory is changed to account for the new location of the file DatabaseHandler.dll
3. Restart and RestartSec: These two settings determine if, and at what interval the service should restart if it stop for whatever reason.
4. SyslogIdentifier: This is what the system log will use to identify the Database Handler, you may change this to whatever you'd like.
5. Environment: These settings must not be changed, the service will not function if these settings are not present, or altered.

4.3 Install

This tells the service who uses this service, this setting should not be changed, otherwise there could be issues with accessing the service.

5 The Configuration.ini File

This file contains the Configurations for the Database Handler, we advise against changing the contents of this file, however, this section gives an explanation of what is in the file, and how changes should be made. **Important: Changing**

the section names within this file will result in the Database Handler not being able to find the settings it needs!

To update any any setting in this file, find the correct "setting" (located on the left hand side of the equals sign) and change the key (located on the right hand side of the equals sign) to the new value. The individual settings are located in "groups", which are in brackets, e.g. "[MySQL Connection]". Below is a discussion of all groups, and the settings contained in each group.

5.1 MySQL Connection

This group tells the Database Handler where the Database is located, how to connect to it, and what username and password to use to connect. This may need to be changed if the username and/or password change.

To update the username and password:

1. Find the setting "user" and change the right hand side of the equals sign to the new username. **Important: As the Database Handler will alter tables; create, update, and delete table entries; and make other modifications to the Database, the user entered here must have root priviledges. See Section 6.1 "Setting up a User for DBH" for more info.**
2. Find the setting "password" and change the right hand side of the equals sign to the password assigned to the user in the "user" setting.

Another setting in this group that may need to be changed is the "DB" setting. This must be the name of the Database schema used by the Database Handler (See Section 6.2 "Setting up the Database" for more info). Lastly, we also have the setting "host" this is the IP address of the server hosting the Database. If the Database is on the same server as the Database Handler, this setting should be *localhost*, otherwise it must be the IPv4 address of the server where it is located.

5.2 MySQL Tables

This group contains basic information about the tables the Database Handler is managing. This information should not be changed unless the MySQL Database table names and/or keys were changed. If any of these were changed:

1. Update grad_plans with the name of the tablke containing the graduation plans
2. Update credentials with the name of the table containing the user credentials of website users
3. Update students_table with the name of the table containing the graduation plans
4. Update degrees_table with the name of the table containing the degrees

5. Update `courses_table` with the name of the table containing the courses
6. Update `catalogs_table` with the name of the table containing the catalogs
7. Update `grad_plans_key` with the name of the column containing the student ID of the student whose graduation plan it is
8. Update `credentials_key` with the name of the column containing the username of website users
9. Update `students_table.key` with the name of the column containing the student ID of a student
10. Update `degrees_table.key` with the name of the column containing the degree ID
11. Update `courses_table.key` with the name of the column containing the course ID
12. Update `catalogs_table.key` with the name of the column containing the catalog ID

5.3 Misc

This section contains the TCP/IP setting used by the website and the Database Handler to communicate, as well as the path to a log file, which the Database Handler will use to log error messages, and other important events.

If it is necessary to update the TCP settings:

1. Update the IP setting with the IP address of where the **webserver** is located. Alternatively, you can update this setting to *Any* which will cause the Database Handler to accept client requests from any IP address, **this is not recommended**.
2. Update the Port setting to the TCP port to accept client requests on. If a new port is chosen, we recommend a port above 10000, ports below this are not guaranteed to be open.

Changing the log path is not recommended, the Database Handler may not have access privileges to the new file, which will cause the Database Handler to shutdown.

6 MySql Setup

This section will explain how to setup, and alter the MySql Database. Throughout this section, we will instruct you to use certain names for the Database Schema, Tables, and Columns, you may replace these with names of your choice; however, it is not recommended to use settings other than the defaults, as any changes made in the Database must also be applied to the Configuration.ini file, which can cause issues if done incorrectly.

6.1 Requirements

This guide assumes that you have already installed MySQL, we recommend you use MariaDB to avoid any issues, but you can use a version of your choice.

Note: The syntax in the scripts, and the instructions herein follow the MariaDB syntax, other MySQL versions may use a slightly different syntax.

In the section 3.2 "Setting up the Database

6.2 Setting up the Database Schema

1. Open a shell, navigate to your home folder, and execute the command:
sudo mysql
You may be required to enter the root password. After this, you should now be in the mysql shell, it should display "mysql >" in your shell.
2. Execute the command: **CREATE SCHEMA IF NOT EXISTS advising_db;**
3. Execute the command: **exit;**
You should now be back in the regular shell
4. Execute the command: **sudo mysql CatalogsTable.sql**
5. Execute the command: **sudo mysql CoursesTable.sql**
6. Execute the command: **sudo mysql CredentialsTable.sql**
7. Execute the command: **sudo mysql DegreesTable.sql**
8. Execute the command: **sudo mysql StudentPlanTable.sql**
9. Execute the command: **sudo mysql StudentssTable.sql**

7 Troubleshooting