

**USER’S**

**MANUAL**

*.NetScaffolder*

**for Users**

Prepared by: Laredo Tirnanic

October, 2018

**Revision Sheet**

|  |  |  |
| --- | --- | --- |
| **Release No.** | **Date** | **Revision Description** |
| Rev. 0 | 10/08/18 | User’s Manual Created |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**USER'S MANUAL**

**TABLE OF CONTENTS**

1.0 GENERAL INFORMATION 1-1

1.1 System Overview 1-1

1.2 Import Sources 1-1

1.3 Data Layers 1-1

1.4 Supported Database Drivers 1-1

2.0 SYSTEM SUMMARY 2-1

2.1 System Configuration 2-1

2.2 User Access Levels 2-1

3.0 GETTING STARTED 3-1

3.1 Setting up MySQL 3-1

3.2 Create the database 3-4

3.3 Setting up database tables 3-4

CREATE TABLE `DSS\_MAIN`.`student\_information` (`DawgTag` INT (9) NOT NULL, `SSN` VARCHAR (11) NOT NULL, `FirstName` CHAR (50), `MiddleInitial` CHAR (2), `LastName` CHAR (50), `Birthdate` VARCHAR (10), `Sex` VARCHAR (6), `Ethnicity` VARCHAR (15), `Status` VARCHAR (13), `MaritalStatus` VARCHAR (9), `Class` VARCHAR (9), `HousingAssistance` VARCHAR (5), `Major` VARCHAR (50), `LocalAddress` VARCHAR (50), `LocalCity` VARCHAR (50), `LocalState` VARCHAR (15), `LocalZip` VARCHAR (10), `LocalPhone` VARCHAR (14), `CellPhone` VARCHAR (14), `HomeAddress` VARCHAR (50), `HomeCity` VARCHAR (50), `HomeState` VARCHAR (15), `HomeZip` VARCHAR (10), `HomePhone` VARCHAR (14), `EmailAddress` VARCHAR (50), `TextConversion` VARCHAR (3), `Wheelchair` VARCHAR (3), `War` VARCHAR (15), `Military` VARCHAR (3), `DHSAffiliation` VARCHAR (3), `DSSWorker` VARCHAR (50), `PrimaryCode` VARCHAR (50), `PrimaryDisability` VARCHAR (100), `SecondaryDisability` VARCHAR (100), `SummerService` VARCHAR (512), `FallService` VARCHAR (512), `SpringService` VARCHAR (512), `FiscalYearService` VARCHAR (512), `Disabled` VARCHAR (3), `CaseStatus` VARCHAR (6), `CaseNotes` LONGTEXT, PRIMARY KEY(`DawgTag`), UNIQUE(`DawgTag`)) TYPE = InnoDB; 3-5

CREATE TABLE `DSS\_MAIN`.`equipment` (`DawgTag` INT (9) NOT NULL, `EquipmentName` VARCHAR (256), `SIUCNumber` INT (10) UNSIGNED, `DateLoaned` VARCHAR (10), `DateDue` VARCHAR (10), `DateReturned` VARCHAR (10), `Notes` VARCHAR (128), `ID` VARCHAR (25) NOT NULL, PRIMARY KEY(`ID`), UNIQUE(`DawgTag`,`ID`)) TYPE = InnoDB; 3-5

3.4 Adding/Removing MySQL Users 3-5

4.0 USING the CLIENT APPLICATION 4-1

4.1 Settings 4-1

4.1.1 Server 4-1

4.1.2 Username 4-1

4.1.3 Password 4-1

4.1.4 Database Name 4-1

4.1.5 Database Table 4-1

4.1.6 Equipment Table 4-1

4.1.7 Save most recent queries 4-1

4.2 Connect/Disconnection to database 4-1

4.3 Student Information 4-1

4.3.1 Add Record 4-2

4.3.2 Update Record 4-2

4.3.3 Clear Fields 4-2

4.4 Equipment Rental 4-2

4.5 Query 4-2

4.5.1 Save Query 4-2

4.5.2 Load Query 4-2

4.5.3 Query Results 4-2

4.5.4 Save Query Results 4-3

4.5.5 Change Query Results Display 4-3

5.0 USING the CONVERSION APPLICATION 5-1

5.1 Connect 5-1

5.2 Convert 5-1

This function converts the Access records to the new MySQL database format. Depending on the amount of records and speed of the computer, this can take several minutes. To assure that the program has not locked up, a progress bar has been implemented to show function progress. 5-1

5.3 Report 5-1

This function should be run after the completion of the convert function. It creates a Notepad text file in the grandparent directory of the conversion application that lists the DawgTag numbers of all student records and equipment records that contain data formats which the conversion application could not convert easily. 5-1

5.4 Disconnect 5-1

Disconnects the conversion application from both the Microsoft Access input database and the MySQL output database. 5-1

**1.0 GENERAL INFORMATION**

# GENERAL INFORMATION

## 1.1 System Overview

.NetScaffolder is a simple, highly flexible scaffolding framework based on T4 Scripts.

The Scaffolder Application is used to import and manage models. Model data can be imported from various sources. These sources can be databases or modelling tool files. Imported data is saved in a metadata file which is then used to generate the different layers.

The layers generated depend on the package selected. Packages consist of several templates. Templates use T4 files and the metadata to generate a layer.

## 1.2 Import Sources

* Edmx Files ( No index data is available )
* MySql Metadata ( Version 5.7 - Known issues with version 8)
* Oracle Metadata
* Sql Server Metadata
* PostgreSQL ( Planned )
* Extendable Custom sources

## 1.3 Data Layers

* Context ( Entity Framework 6.0 / Entity Framework Core / NHibernate)
* Entities
* Repository Layer
* Application Service
* Dto

## 1.4 Supported Database Drivers

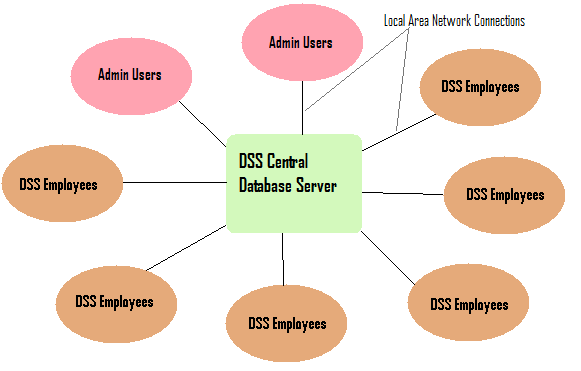
* Entity Famework 6.0 ( MySql, Sql Server, Oracle, PostgreSQL )
* Entity Framework Core 2.1 ( MySql - Pomelo Driver, Sql Server )
* Adding support for NHibernate ( Mysql, Sql Server, Oracle )

**2.0 SYSTEM SUMMARY**

# SYSTEM SUMMARY

## 2.1 System Configuration

Each employee uses their own computer running Windows OS. On those computers is the client application that will connect to a central database running Windows and MySQL database software. All DSS computers are connected on a LAN with no outside connection to the internet. This is because sensitive information such as social security numbers are being sent across the network.



## 2.2 User Access Levels

There are two different types of users as depicted in the image above. Admin Users are DSS employees who have authority to view and change student’s information within the database. DSS Employees refer to employees who have permission to read student’s records, but not change any of them. The Admin Users are still DSS employees, but are part of the administration of DSS. When we refer to DSS employees as users with limited permissions we are typically referring to student workers who are working for DSS.

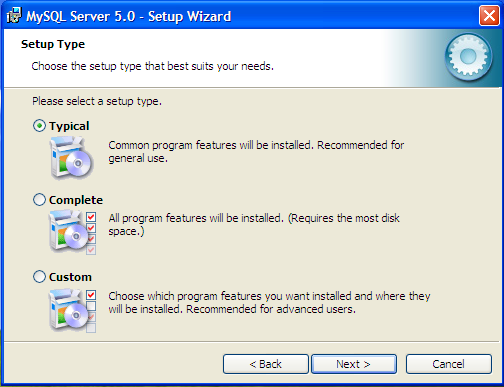
**3.0 GETTING STARTED**

# GETTING STARTED

## 3.1 Setting up MySQL

We first start by running “MySql-setup.exe”. This will begin the installation of MySQL.

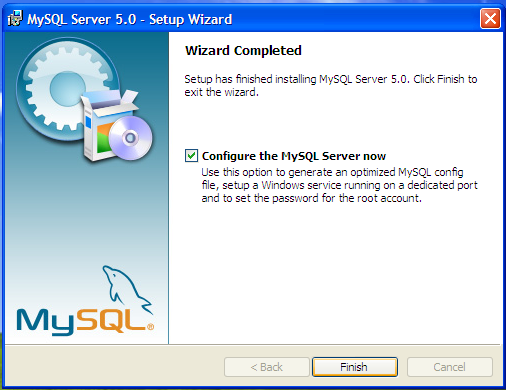
-Select “Typical” installation then click the “Next >” button



-Click the “Install” button to begin installation

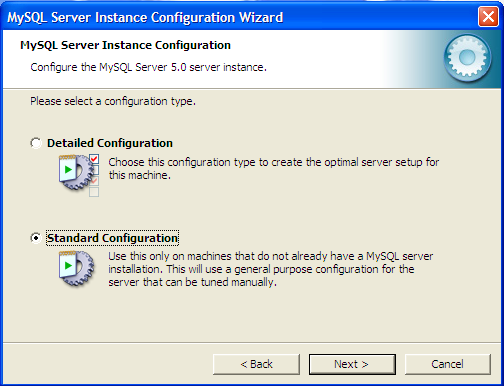
-Once the installation completes click “Next >” twice

-Now insure “Configure the MySQL Server now” check box is selected and press “Finish”

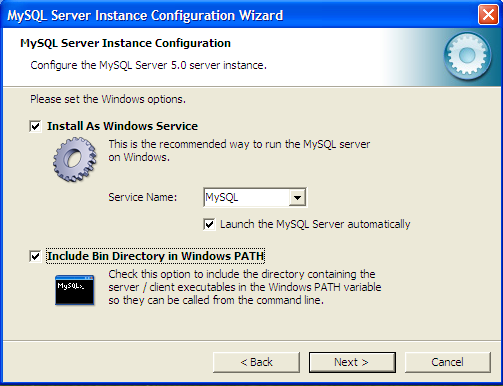


-Click “Next >” on the Configuration Wizard to begin the configuration

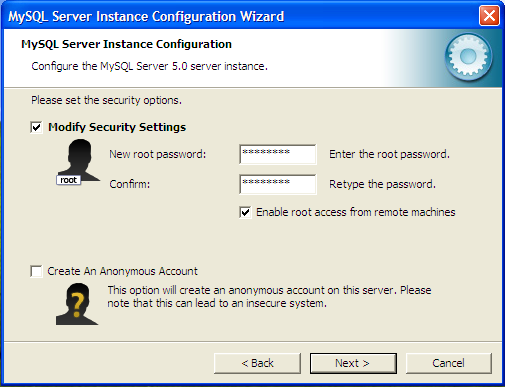
-Select the radio button “Standard Configuration” and click “Next >”



-Ensure “Install As Windows Service” checkbox is checked, the service name is “MySQL” and “Launch the MySQL Server automatically” is checked. Also make sure “Include Bin Directory in Windows PATH” checkbox is checked. Click “Next >” to continue



-Now make sure “Modify Security Settings” and “Enable root access from remote machines” are the only two check boxes selected. Input the desired password that only the administrator will know. This password will be used for adding and removing MySQL users as well as setting up the database tables. The username for the root account is “root” and this account should NOT be used to log in from the DSS Database Suite client program. DSS employees should use the accounts created in 3.3 as their login accounts. Click “Next >” then “Execute” to setup the MySQL server with the specified settings.



-Click “Finish” once the processing is complete.

After the installation is complete you will need to use the MySQL Command Line Client (which can be found under Start Menu -> MySQL -> MySQL Server 5.0) Once the command prompt comes up you will need to enter the root password you set in the MySQL setup. From this command line you will be able to complete setting up the database tables and adding MySQL Users

## 3.2 Create the database

The database is like a container for all of the information used in this application. We need to first add a database and in this example we will use DSS\_MAIN as the name of the database, but anything can be used. To create the database we use the following command:

CREATE DATABASE `DSS\_MAIN`;

Note that the ticks surrounding the work DSS\_MAIN are the ticks located to the left of the number one (1) on a standard QEWRTY keyboard.

## 3.3 Setting up database tables

Once the database has been created we need to add tables to it. Database tables are used to store information for every student. They can be thought of as spreadsheets where each row is a record that corresponds to a specific student or piece of equipment. The database uses only two different tables to operate. We need one table for all of the student’s information and another table for all equipment used by students. To create the tables the following commands should be used.

This first command creates the table for student information. The name DSS\_MAIN should be changed to whatever you named the database in step 3.2 and the name student\_information can be changed to the desired table name for the student’s information.

## CREATE TABLE `DSS\_MAIN`.`student\_information` (`DawgTag` INT (9) NOT NULL, `SSN` VARCHAR (11) NOT NULL, `FirstName` CHAR (50), `MiddleInitial` CHAR (2), `LastName` CHAR (50), `Birthdate` VARCHAR (10), `Sex` VARCHAR (6), `Ethnicity` VARCHAR (15), `Status` VARCHAR (13), `MaritalStatus` VARCHAR (9), `Class` VARCHAR (9), `HousingAssistance` VARCHAR (5), `Major` VARCHAR (50), `LocalAddress` VARCHAR (50), `LocalCity` VARCHAR (50), `LocalState` VARCHAR (15), `LocalZip` VARCHAR (10), `LocalPhone` VARCHAR (14), `CellPhone` VARCHAR (14), `HomeAddress` VARCHAR (50), `HomeCity` VARCHAR (50), `HomeState` VARCHAR (15), `HomeZip` VARCHAR (10), `HomePhone` VARCHAR (14), `EmailAddress` VARCHAR (50), `TextConversion` VARCHAR (3), `Wheelchair` VARCHAR (3), `War` VARCHAR (15), `Military` VARCHAR (3), `DHSAffiliation` VARCHAR (3), `DSSWorker` VARCHAR (50), `PrimaryCode` VARCHAR (50), `PrimaryDisability` VARCHAR (100), `SecondaryDisability` VARCHAR (100), `SummerService` VARCHAR (512), `FallService` VARCHAR (512), `SpringService` VARCHAR (512), `FiscalYearService` VARCHAR (512), `Disabled` VARCHAR (3), `CaseStatus` VARCHAR (6), `CaseNotes` LONGTEXT, PRIMARY KEY(`DawgTag`), UNIQUE(`DawgTag`)) TYPE = InnoDB;

The next command is used to create the equipment table. Once again you must change the name DSS\_MAIN to whatever was named in step 3.2 and may change the table name of equipment to whatever is desired. The command is this:

## CREATE TABLE `DSS\_MAIN`.`equipment` (`DawgTag` INT (9) NOT NULL, `EquipmentName` VARCHAR (256), `SIUCNumber` INT (10) UNSIGNED, `DateLoaned` VARCHAR (10), `DateDue` VARCHAR (10), `DateReturned` VARCHAR (10), `Notes` VARCHAR (128), `ID` VARCHAR (25) NOT NULL, PRIMARY KEY(`ID`), UNIQUE(`DawgTag`,`ID`)) TYPE = InnoDB;

Note that the only fields that can be changed in the previous two commands are the two words immediately after the word TABLE. In this case DSS\_MAIN and wither student\_information or equipment. The other fields are the names of the columns that are necessary to the function of the client program.

## 3.4 Adding/Removing MySQL Users

The next step in the setup process is for the Administrator to setup the users who will be using the database. Based on the two types of users described above we need to either create users who have full privileges to the database, or limited (read-only) privileges. To create use the following command:

CREATE USER ‘user\_name’@’%’ IDENTIFIED BY ‘password\_for\_new\_user’;

After the user is created you must set privileges for that user. If the user needs full access privileges to the database use this command where the only thing that needs to be changed is the word new\_username which should be the name of the user just created:

GRANT ALL ON \*.\* TO ‘new\_username’@’%’;

If the new user is a user with less privileges and can only query the database the following command should be used where the only thing that needs changed is the word new\_username which should be the name of the user just created:

GRANT SELECT ON \*.\* TO ‘new\_username’@’%’;

To remove a user use the following command where the word username should be replaced with the name of the user you want to delete:

DROP USER ‘username’@’%’;

**3.5** **Install .NET Framework 2.0 or later**

If the computer already has the .NET Framework installed this step can be skipped, otherwise run the “dot NET Framework.exe” installer. This will install version 2. If you already have a later version of .NET Framework you do not need to worry about this step.

**3.6** **Installing the MySQL Connector**

MySQL Connector is a library that links the DSS Database Suite to the MySQL Database to allow communication and transfer of data between the two. To install this run “MySQL.Data.exe”

-Once the installation screen appears on the screen click “Next” to begin the installation.

-Click “Typical” as the type of installation then click “Install” to begin.

-When the installation has completed click “Finish”

**3.7** **Installing the DSS Database Suite Client**

The Database Suite doesn’t actually need installing. All you need to do is place the program “DatabaseSuite.exe” into a convenient location. The Client itself is only one standalone program, but to get it to work with the MySQL database properly you had to perform the previous steps.

**4.0 USING THE CLIENT APPLICATION**

# USING the CLIENT APPLICATION

## 4.1 Settings

Changing the settings to fit the current user is the first step that should be taken by a new user to the computer before using the application. The settings can be access from the “Settings” menu in the main application window. There are seven (7) settings that can be changed to fit the current user’s needs. When the settings are changed, the changes will be saved even when the program is restarted. Users may need to contact the network administrator to find out all of the details for these settings.

### 4.1.1 Server

The IP address of the computer on the LAN that is running the MySQL server. You may either type the IP address of the database server or the network name of the computer

### 4.1.2 Username

The name of the current user that was setup in the MySQL server by the Administrator specifically for the current user

### 4.1.3 Password

The password that corresponds to the current user that was setup in the MySQL server by the Administrator specifically for the current user

### 4.1.4 Database Name

The name of the database specified in section 3.2 when MySQL was being setup

### 4.1.5 Database Table

The name of the student information table that was added to the database in section 3.3 when MySQL was being setup

### 4.1.6 Equipment Table

The name of the equipment table that was added to the database in section 3.3 when MySQL was being setup

### 4.1.7 Save most recent queries

If this box is checked queries that are made will be saved so that the next time the user returns to the query form, the previous credentials will still be displayed. If it is not checked a new query form will be blank.

## 4.2 Connect/Disconnection to database

To be able to access the database the user must connect to it through the “Connect” menu found under the “Database” menu. When the user is finished with their task they can disconnect from the database using the “Disconnect” menu under the “Database” menu.

## 4.3 Student Information

Student information is stored in three (3) parts. The first is the basic personal information, contact information and services provided which all three wrapped up into one is also called student information. The second is Equipment Rental. This tab is where information is stored regarding any equipment the student might have rented from the university. The final is Case Notes. Case Notes are any kind of notes that will help the user better understand the student’s case. Once the appropriate information has been filled out the student records can be added to the database or edited via the function described in this section.

### 4.3.1 Add Record

Once all of the data fields are filled in on the main application window (i.e. Student Information, Equipment Rental, and Case Notes) the record can be added to the database from the “Add” menu under the “Student” menu on the main form.

### 4.3.2 Update Record

After a record has been loaded and information has been changed the user can update the record by clicking on “Update” under the “Student” menu.

### 4.3.3 Clear Fields

This is a convenience feature that will clear all fields in the main form. When the user selects “Clear Fields” under the “Student” menu, they will be prompted to confirm the clear. If they user selects “yes” the information will be cleared, otherwise nothing will happen.

## 4.4 Equipment Rental

Once a record has been loaded into the main form the user can add, edit, or remove equipment entries from the database for that particular student. This can be done one of two ways: the menu items under the “Equipment” menu, or by right clicking in the Equipment Rental list box. To edit and remove records the user must have a record selected, however to add a record the user does not.

## 4.5 Query

To be able to search the database for a particular student, or a group of students that meet a certain criteria, the user must select “Query” from the “Student” menu. This will bring up a new dialog box with all of the same fields that are on the main form. To search the database type in the desired criteria in the fields on the query form then click the “OK” button on the bottom left of the form. Once this is done a new dialog box will come up showing the results.

### 4.5.1 Save Query

What this does is save the criteria that are currently entered in the query form into a file on the hard disc specified by the user. This allows common queries to be saved so the user does not have to type in the same information every time. To save a query the user must click the “Save” button on the query form after the desired query criteria has been filled in.

### 4.5.2 Load Query

To retrieve the search criteria from a saved query the user will select “Load” from the query form, then choose the file name and location of the query to load.

### 4.5.3 Query Results

Once the search criteria have been entered and the user clicks “OK” on the query form all of the results that match the search will be shown in the query results form. From here the user can either double click, or single click then click the “OK” button to load the record data into the main form.

### 4.5.4 Save Query Results

Once a query has been performed and the user can choose to save the query results. To do so they will click the “Save” button on the query results form. Saving the query results will save all of the information for each student that was successfully found based on that query (i.e. all of the students in the query results). This data is save in a Microsoft Excel compatible XML format and can be further edited to remove unwanted columns via Excel.

### 4.5.5 Change Query Results Display

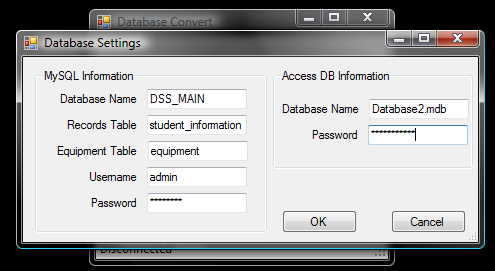
In the query results form the user can select which columns they want displayed by right clicking in the list view and selecting the menu item corresponding to the field they want to see.

**5.0 USING THE CONVERSION APPLICATION**

# USING the CONVERSION APPLICATION

## 5.1 Connect

Connects the conversion application to the existing Microsoft Access input database and MySQL output database. A settings form loads when connect is clicked to allow user input of database information. There are (7) settings that can be changed to fit the user’s needs. Settings must be input every time the application is run, but the application is only intended to be run once. This software must be run by an administrative user, so as to allow the application to create all the necessary records in the MySQL database. To facilitate Access database conversion, a copy of the Access database must be inserted into the parent directory of the conversion application, and the input file should be listed as ‘xxxxxx.mdb’.



## 5.2 Convert

### This function converts the Access records to the new MySQL database format. Depending on the amount of records and speed of the computer, this can take several minutes. To assure that the program has not locked up, a progress bar has been implemented to show function progress.

## 5.3 Report

### This function should be run after the completion of the convert function. It creates a Notepad text file in the grandparent directory of the conversion application that lists the DawgTag numbers of all student records and equipment records that contain data formats which the conversion application could not convert easily.

## 5.4 Disconnect

### Disconnects the conversion application from both the Microsoft Access input database and the MySQL output database.