

## **Introduction**

wx2sql is an application that processes weather observations recorded by WeatherLink for entry into a MYSQL database. To ensure cross-platform functionality, wx2sql was developed in Java using JavaFX. wx2sql will run, and has been tested on, Windows, Linux, and macOS. However, only a Windows installer is available for download.

With WeatherLink, the folks at Davis provide a file layout document that was essential to developing wx2sql. Also, their support team was very willing to answer a couple of questions that I had in regards to the file layout. I want to say thank you to these guys. This application would not have been possible without their openness. In addition to the quality of their products, this type of support and openness is what will make me a repeat customer.

## **\*Important Bug Fix\***

Versions of wx2sql prior to 0.1.4 contained a bug that affected the maximum wind speed and hi wind speed values that could be inserted into the database. The bug caused values that were above 30 mph to be inserted into the database as a null value.

To correct the issue, data files containing wind speed values above 30mph will need to be reprocessed.

## **Disclaimer**

**This program comes as-is, with no warranties either expressed or implied. I am not, and cannot, be held responsible for any damages or loss of data that occurs as a result of downloading and using wx2sql or its associated programs/scripts. Only continue if you completely understand this, accept these risks, have read and understand the known issues listed below, and have the knowledge and skills necessary to implement this process.**

Required Components.....	3
Upgrading from Previous Versions.....	3
Installing wx2sql .....	3
Installed Files .....	3
Launching wx2sql – Configuration File .....	4
Configure Application Settings .....	4
Configure Application Settings – Logging ( <i>continued</i> ) .....	6
Upgrade Database ( <i>continued</i> ).....	8
Upgrade Database ( <i>continued</i> ).....	9
WLK File Utilities .....	9
WLK File Utilities – Data Export ( <i>continued</i> ).....	10
WLK File Utilities – Health Check ( <i>continued</i> ).....	11
Job Types and Queues .....	11
Job Types and Queues ( <i>continued</i> ) .....	12
Job Types and Queues ( <i>continued</i> ) .....	13
Understanding the Database.....	13
Error Correction .....	14
Error Correction ( <i>continued</i> ) .....	15
Error Codes .....	15
0.1.1 Changelog .....	16
0.1.0 Final Changelog .....	17
0.1.0 Beta 3 Changelog.....	17
0.1.0 Beta 2 Changelog.....	17
Planned Improvements .....	18

## Required Components

To get a working installation of wx2sql, you will need the following:

- a working instance of MYSQL
- wx2sql setup executable, available for download from <http://www.hoosierweather.com/software>
- Java 8

## Upgrading from Previous Versions

Version 0.1.3 contains changes to the database schema from version 0.1.2. After upgrading, users are required to update the database prior to the processing of observational data. Please refer to the **Database** section for detailed instructions on completing necessary database upgrades. Please refer to the **0.1.3 Changelog** section for a detailed list of enhancements.

## Installing wx2sql

Once downloaded, launch the wx2sql\_setup.exe. I won't spend a lot of time on this, it really is as easy as 'Next', 'Install', and 'Finish'. Please see **Figure 3-1** thru **Figure 3-3**. wx2sql is a Java based application and has been tested on Linux, macOS, and Windows. However, as of the date of this documentation, only the Windows installer is available. If there is enough interest, I will look at packaging this on other platforms.

Figure 3-1

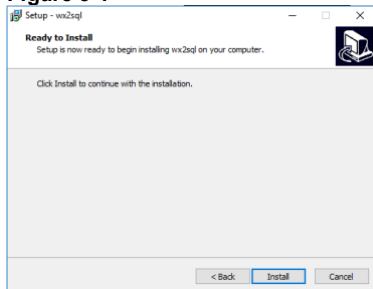


Figure 3-2

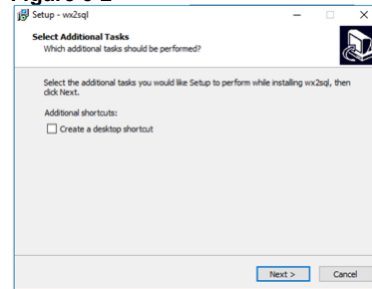
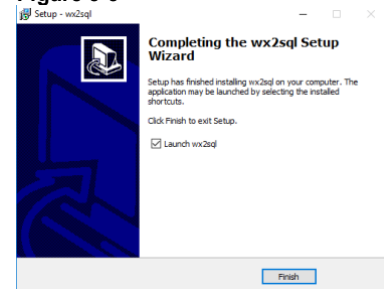


Figure 3-3



## Installed Files

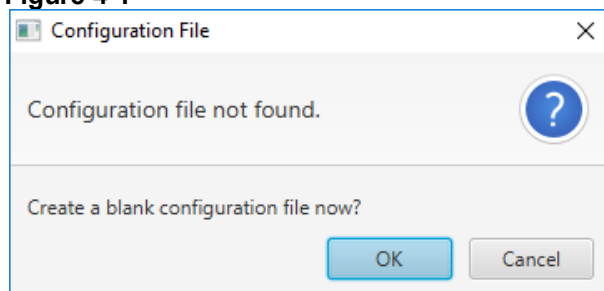
During installation, the following directories and files will be created on your system:

- C:\Users\<USERNAME>\wx2sql
  - This directory contains two very important files, **config.properties** and **jobQueue.xml**
    - **config.properties** This file contains all of the configuration options that the user sets from Preferences. It is highly recommended that you do not make changes to this file manually and that all changes are made from Preferences within wx2sql.
    - **jobQueue.xml** This XML file is used to store the progress of all job queues. This file is updated constantly while the application is running. During the shutdown process, the last task of a job is to record its current status within this file before going to sleep. It is highly recommended that you do not manually modify this file. Doing so can cause unexpected behavior in the application
- C:\Users\<USERNAME>\wx2sql\diagReports
  - Contains pdf reports that are generated when a health check is performed on an observation file. The reports in this directory will be listed in the WLK File Utilities Tab and can be removed from there. Removing reports manually through this file is also permissible and will not cause any issues with the application
- C:\Program Files (x86)\wx2sql
  - Contains the wx2sql executable file and other necessary files necessary to uninstall the application
- C:\Program Files (x86)\wx2sql\lib
  - Contains various .jar files, and necessary license files, needed by wx2sql in order to run.

## Launching wx2sql – Configuration File

If you have installed wx2sql for the first time, and are launching the application, wx2sql will discover that a configuration file is not present. A prompt will be displayed, giving the user the option to create a blank configuration file. To allow wx2sql to create the configuration file, select **OK**. Please reference **Figure 4-1**.

**Figure 4-1**



With each new release, it is possible that configuration options will be added, removed, or changed. These modifications are handled automatically by wx2sql. If changes are necessary, wx2sql will convert the existing configuration parameters when the application is launched. No user interaction is required.

## Configure Application Settings

After installation, configuring wx2sql is your next step. Let's begin by setting the application preferences. Select **Edit → Preferences**.

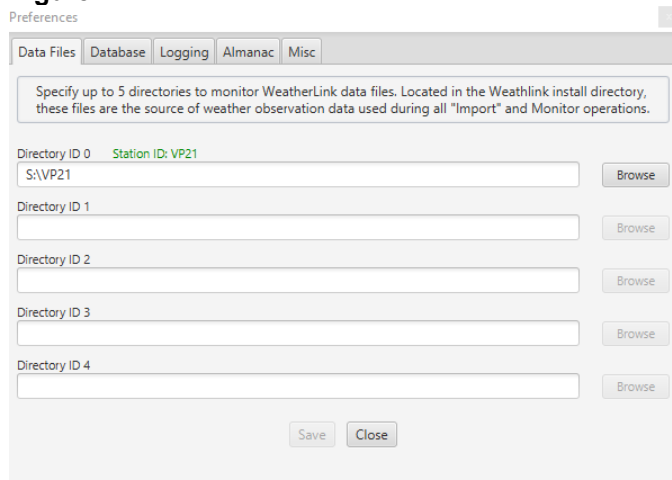
### Preferences Tab - Data Files

The first configuration option that will be addressed is the location of the WeatherLink data files. By default, these files are located in the WeatherLink install directory, in a sub-directory that is the name of the weather station. Only data files in the directory specified here will be available to Import or Monitor jobs. To specify the desired directory, select **Browse**, then navigate to the directory where these files reside. Select **Save** to write changes to the configuration file. In **Figure 4-1**, you can see additional fields are visible to specify multiple source directories. In version 0.1.3, these fields are disabled. They will be enabled in future versions.

#### **\*\*Important\*\***

With support for multiple weather stations coming soon, a way had to be found to identify which weather station the observations belong. This information is not recorded in the data file itself. Therefore, we determine the weather station name by the directory in which the files reside. For example, if **c:\WeatherLink\WeatherStation1** is the path, **Weather Station1** will be the weather station name. In **Figure 4-1**, you will see that the Station ID is listed in Green to help clarify this.

**Figure 4-1**



## Preferences Tab - Database

wx2sql utilizes two databases, wx2sql\_prod and wx2sql\_test. These database names are hard coded and cannot be changed. However, some hosting providers require customers have a prefix to each database name. If you are required to do this, select **Use Database Name Prefix** then specify the necessary prefix. Please reference **Figure 5-1**.

Connecting to the database will require four pieces of information:

- Database
  - This dropdown will allow you to select the Production or Test Database. You will set the credentials for the database you have selected here. For example, to specify credentials for the production database, select “wx2sql\_prod” to specify credentials for the production database. Select “wx2sql\_test” to specify the credentials for the test database. It is important to note that the database selected here is also considered the “Active Database”. All database operations will be performed on this database.
- Database Username
- Database Password
- Hostname/IP Address of the MYSQL server

By default, Database Passwords are stored in clear text within the configuration. To store the password as an encrypted value, select the option **Store Password Securely**.

When these configured fields are populated, the **Test Connection** button will be enabled. After making changes to the connection settings, please test your connection. If your connection is successful, the server version and Database Schema version will be displayed. Please see **Figure 5-1**

**Figure 5-1**

The screenshot shows the 'Preferences' window with the 'Database' tab selected. The window has a title bar 'Preferences' and a tab bar with 'Data Files', 'Database', 'Logging', 'Almanac', and 'Misc'. The 'Database' tab contains a text box with instructions: 'Specify the information necessary to connect to the wx2sql database. You will need the username, password, IP address/hostname, and the database type. With this information provided, make sure to test the connection.' Below this are several fields: a checkbox for 'Use Database Name Prefix' (unchecked), a 'Database Name Prefix' text field, a 'Database' dropdown menu (set to 'wx2sql\_test'), a 'Username' text field (set to 'wx2sql\_test'), a 'Password' text field (masked with dots), a checkbox for 'Store Password Securely' (unchecked), and a 'Hostname' text field (set to '107.180.56.148'). To the right of these fields is a 'Database Test Results' section with 'Version' and 'Schema' text fields. At the bottom are 'Save', 'Close', and 'Test Connection' buttons.

## Preferences Tab - Logging

The logging subsystem of wx2sql manages log events generated by the application and will display these log entries on the application’s main window in the Log Viewer. There are two configurable settings here, Log Retention and Log Level.

- Log Retention allows you to specify the number of log entries that you want the application to keep. You can select **50, 100, 250, 500, or 1000**. Once the number of logs reaches the maximum level, the oldest log entries will be removed.
- Log level determines how verbose you want the logging to be. For day-to-day operations, **Normal** is sufficient. **Debug** can be used to increase the verbosity of logging and will aid in troubleshooting.

Please see **Figure 6-1**.

## Configure Application Settings – Logging *(continued)*

**Figure 6-1**

Figure 6-1 shows the 'Logging' tab of the 'Preferences' dialog box. The dialog has tabs for 'Data Files', 'Database', 'Logging', 'Almanac', and 'Misc'. A text box at the top explains the logging level: 'Specify the desired logging level of wx2sql. The default setting of "Normal" is fine for day to day operations and is the recommended setting. If you require more verbose output on the program operations, for troubleshooting purposes, set the logging level to "Debug".' Below this, a label says 'Select the number of log entries you would like to keep.' followed by a 'Select Retention Setting' dropdown menu. Then, a label says 'Select the Log Level' with two radio buttons: 'Normal' (selected) and 'Debug'. At the bottom are 'Save' and 'Close' buttons.

### Preferences Tab - Almanac

wx2sql can connect to the United States Naval Observatory in order to pull Moon Phase, moonrise and moonset, and sunrise and sunset. The following information is required for this operation:

#### **City/Town**

Rather self-explanatory, but this is the city for which you are going to pull information for.

#### **State**

Dropdown the list of states and select the state in which your city is located.

#### **Time of Data Pull**

Pulling information from the USNO will occur once a day. This can happen at 00:00, 06:00, 12:00, or 18:00. Select your desired time.

#### **Days to Pull in Advance**

All of the previous pulled data will remain in the database, you can configure how many days in advance for which you want to pull data.

Once you have configured all of the options, it is highly recommended that you select the **Test Pull** button. This connects to the USNO web service, validating your settings. If the City/Town and State combination you entered cannot be found, an error will be displayed. Please see **Figure 6-2**

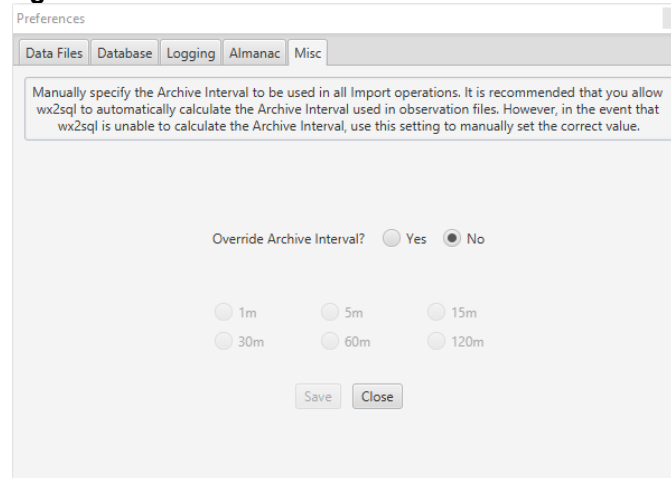
**Figure 6-2**

Figure 6-2 shows the 'Almanac' tab of the 'Preferences' dialog box. The dialog has tabs for 'Data Files', 'Database', 'Logging', 'Almanac', and 'Misc'. A text box at the top explains the data pull: 'Based on the location specified, wx2sql will connect to the United States Naval Observatory to pull in Sun and Moon rise/set information. This information will be stored in the wx2sql database.' Below this, there are three sections: 'City/Town' with a text input field, 'State' with a 'Choose State' dropdown menu, and 'Time of Data Pull' with four radio buttons: '00:00', '06:00', '12:00', and '18:00'. A red error message 'You must select a time to perform data pulls.' is displayed below the time selection. To the right, there is a 'Days to Pull in Advance' section with three radio buttons: '15 days', '30 days', and '45 days', and a red error message 'You must select a value'. At the bottom are 'Save', 'Close', and 'Test Pull' buttons.

## Preferences Tab - Misc

The final configurations are made on the Misc tab. As part of wx2sql's error correction model, it will attempt to automatically calculate what the Archive Interval should be for the day being processed. If this value is incorrectly calculated, it may prevent a file from being imported. If the Archive Interval is calculated incorrectly, you can choose to override the calculated value by changing the **Override** setting to "Yes", then selecting what value the Archive Interval should be. Please see **Figure 7-1**.

**Figure 7-1**



## Setting up the Database

Creation of database tables will be covered later. In preparation of installing wx2sql, you will need to create two empty databases and the associated users.. It is outside of the scope of this document to provide a step-by-step guide to assist with the setup and configuration of a MYSQL server. There is plenty of good information readily available with a quick Google search. As we continue, we will operate under the assumption that you have a working instance of MYSQL.

Please ensure the following steps have been completed.

- Create two databases, wx2sql\_prod and wx2sql\_test. One database will be used for 'production' data, while the other can be used for any testing you may want to do without affecting your production data.
- Create a user for each of these databases. Document the usernames and passwords. This information will be used later during the configuration of wx2sql. Each user will need the following permissions:
  - ALTER
  - INSERT
  - SELECT
  - UPDATE
  -

## Creating Tables

Creating database tables is no longer accomplished through the execution of an external SQL script. To create database tables, select the **Database** section for either the production or test database from the overview screen. Select the **Database Task** dropdown and select **Create Database Tables**. To acknowledge this is your intent, select **I Understand** and then select **Execute**. Please reference **Figure 8-1**

## Deleting Tables

To delete database tables, you are no longer required to do so from the MYSQL server directly. To delete database tables, select the **Database** section for either the production or test database from the overview screen. Select the **Database Task** dropdown and select **Delete Database Tables**. To acknowledge this is your intent, select **I Understand** and then select **Execute**. Please reference **Figure 8-2**.

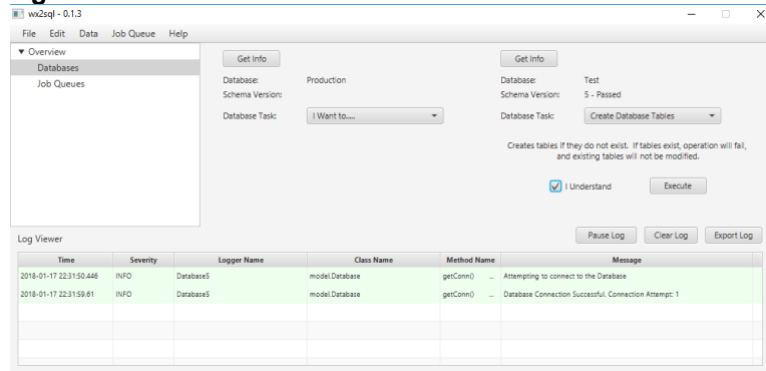
## Upgrade Database

Upgrading the database schema is no longer accomplished through the execution of an external SQL script. If a database upgrade is required, you can perform the upgrade selecting the **Database** section for either the production or test database from the overview screen. Select the **Database Task** dropdown and select **Upgrade Database**. To acknowledge this is your intent, select **I Understand** and then select **Execute**. Please reference **Figure 8-3**

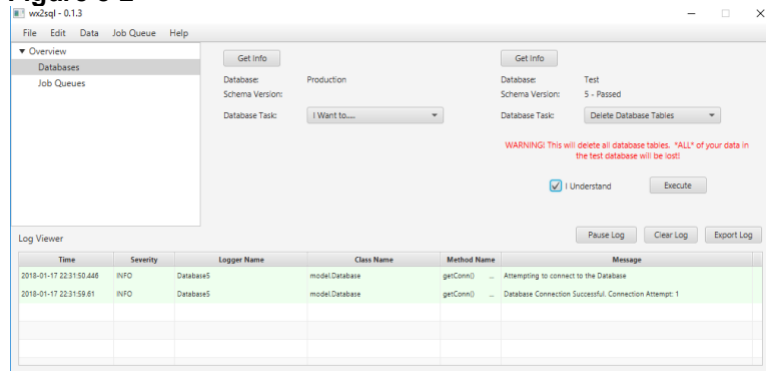
## Upgrade Database *(continued)*

The upgrade screen will then display. On this screen, the database will be evaluated to determine if an upgrade is required. If required, an explanation of the upgrade will be displayed. If information is required for the upgrade, the necessary workflow will be displayed. Once any required workflow is completed, the **Upgrade** button will be enabled. To proceed with the upgrade, select the **Upgrade** button. Please reference **Figure 9-4**

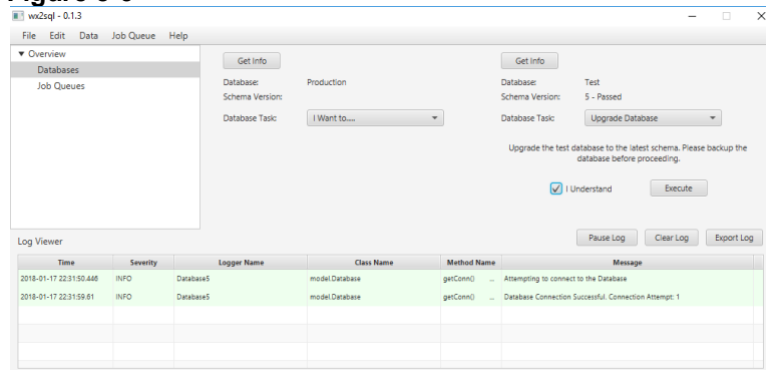
**Figure 8-1**



**Figure 8-2**



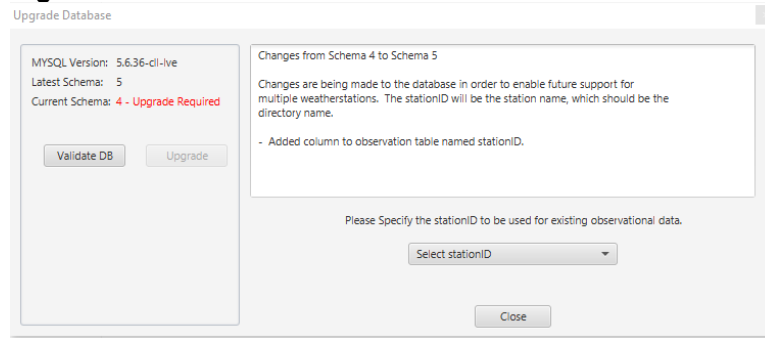
**Figure 8-3**





## Upgrade Database *(continued)*

**Figure 8-4**



### WLK File Utilities

Currently, two file utilities are available in wx2sql: Data Export and Health check.

#### WLK File Utilities – Data Export

Data export is a utility that will read \*.wlk files, interpret the data, and export the observations to a \*.csv file. Several configuration options are available.

##### **Data Source** \*required

Select the format of the data source. Currently, the only option is a WeatherLink Data File.

##### **Format** \*required

Select the format of the output file. Currently, the only option is to export data in a csv format.

##### **Raw Output**

Selecting this option will output data directly from the data file with no formatting applied.

##### **Translate Observation Time**

Observation time data stored in the WeatherLink data file is stored in a format called 'Packed Time'. Packed Time is the number of minutes since midnight. Selecting this option will output the time in hh:mm a (12-hour) format. If enabled, additional options are enabled: **Display as Full Timestamp**, **Translate Other Data**, and **UTC Offset**.

##### **Translate Other Data**

This option will take the raw observational data and output it in a normal format. For example, a temperature of '34.9' is stored in the file as '349'. This option will convert the data to a 'real' reading and output as '34.9'.

##### **Display as Full Timestamp**

This option will translate the Packed Time of the data file into a timestamp with MM/DD/YY HH:mm (24 hr) format.

##### **UTC Format**

With **Display as Full Timestamp** enabled, this option allows you to convert the native time of the observation to a different timezone.

##### **Source File**

Select **Browse** and open the file for which you would like to export data from.

##### **Output File**

Select **Browse** to select the destination directory and filename of the output file.

##### **Data Start Date**

If you wish to only export a subset of data from the file, selecting this option will let you select a valid start date from the file. For example, you can elect to export data beginning on the 15<sup>th</sup> of the month instead of the 1<sup>st</sup>.

## WLK File Utilities – Data Export *(continued)*

### Data End Date

Like the start date, you can elect to change the last day for which you can export data.

Please reference **Figure 10-1**.

**Figure 10-1**

The screenshot shows the 'WLK File Utilities' application window with the 'Data Export' tab selected. The 'Data Source' is set to 'WeatherLink Data File'. The 'Format' is 'Comma-Separated Value (.csv)'. The 'Source File' is 'S:\data\_files\2008-04.wlk' and the 'Output File' is 'C:\Users\mike\untitled.csv'. The 'File Version' is 'WDATS.3' and the 'Observation Count' is '24762'. The 'Data Start Date' is '4/13/2008' and the 'Data End Date' is '4/30/2008'. The 'Display as Full Timestamp' checkbox is checked. The 'UTC Offset' is '-07:00'. The 'Include CSV Column Header' and 'Include WLK File Header Info' checkboxes are checked. The 'Export' and 'Close' buttons are at the bottom.

## WLK File Utilities – Health Check

As the first tool in a planned suite of WLK file utilities, Health Check enables you to analyze your existing WLK files, checking for a variety of potential issues. Health Check verifies the following items:

- Analyzes each day to ensure it does not contain more than the maximum allowed number of observations.
- Verifies each data record's record type to ensure that it is of a valid type.
- Analyzed the data file to detect any duplicate observations.
- Verifies each data record's record type to ensure that it is valid.
- Analyze the data file to detect any duplicate observations.
- Ensure the number of total observations in the file header matches the number of actual observations contained in the file.

### Generate the Report

To analyze a file and generate a summary report, select **Data → File Utilities**. On the **Health Check** tab, click the **Browse** button. Using the file selector tool, browse to then select the data file you wish to analyze. Finally, click the **Run Diagnostics** button. Once complete, you will see the report listed in the table.

### Contents of the Report

The top portion of the report contains information gathered from the Header section of the data file. This includes the file version, total observations, days with observations, and the result of the diagnostic. **Please see figure 11-1.**

## WLK File Utilities – Health Check *(continued)*

Figure 11-1

### WLK File Health Check

Report Created on 04/13/2017 00:14:43 EDT

Data File:	OLD BROKEN2017-02.wlk	Total Observations:	38753	<b>Result: Failed</b>
File Version:	WDAT5.3	Days with Observations:	26	

Below the header summary, you will find descriptions of the issues, if any, that were found. **Please see figure 11-2.**

Figure 11-2

#### Errors Found: 3

- Number of observations for day 26 exceed maximum value (1440). Found 2706 observations.
- Invalid Data Summary Record Type Found: Type 0 found on day 26 at index 1280
- 2703 Duplicate records found

The rest of the document displays data summaries for each day observations were found. If the utility found a problem for a particular day, that day will be displayed in red. **Please see figure 11-3.**

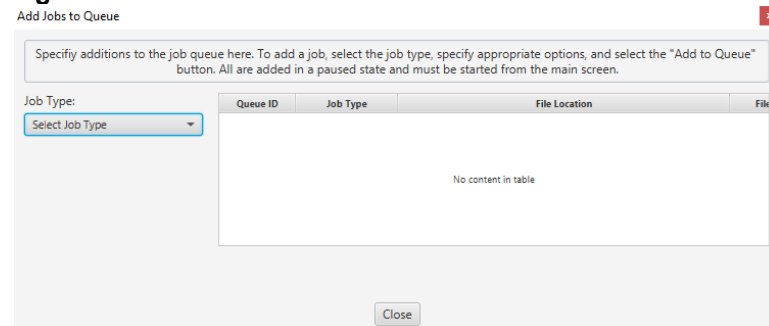
Figure 11-3

17	1442	Yes(1)	Yes(1)	1440	1442	false
18	1442	Yes(1)	Yes(1)	1440	1442	false
19	1441	Yes(1)	Yes(1)	1439	1441	false
20	1442	Yes(1)	Yes(1)	1440	1442	false
21	1442	Yes(1)	Yes(1)	1440	1442	false
22	1442	Yes(1)	Yes(1)	1440	1442	false
23	1441	Yes(1)	Yes(1)	1439	1441	false
24	1442	Yes(1)	Yes(1)	1440	1442	false
25	1442	Yes(1)	Yes(1)	1440	1442	false
26	2706	Yes(1)	Yes(1)	2703	2705	false

## Job Types and Queues

Each job type, when running, is assigned to a Job Queue. Wx2sql provides you with up to five queues in which to run your jobs. You can run any combination of the available job types: Almanac Sync, Import, and Monitor. Jobs are added to the queue from the **Add Jobs to Queue** window. This can be accessed by selecting **Job Queue → Add Jobs to Queue**. The various job types each have options that are specific to the job, these are explained in more detail below. All jobs are added in the “Paused” state. Once the job is added, close the window to get back to the wx2sql main window. To start a job, right-click on the Job and Select **Resume**. In addition, by right-clicking on the job, a job can be **Canceled** and then removed from the queue all together. **Please see Figure 11-4.**

Figure 11-4



### Job Type: Almanac Sync

The purpose of the purpose of the Almanac Sync job type is to download information from the USNO web service and input that data into our database. Once you have selected **Almanac Sync** for the job type, you will be presented with two more options: **Delete Existing Data** and **Perform Sync on Job Start**. **Delete Existing Data**, if selected, will remove all Almanac information from the database before attempting to pull in new data. This is done only once,

## Job Types and Queues *(continued)*

and only when the job is newly added and started. If you wish to perform this task later, you will have to remove any existing Almanac Sync job and re-add it with this option selected. **Perform Sync on Job Start** will, as soon as the job is started, begin pulling in the amount of data that was specified in the Preferences section. Once complete, the job will schedule the next pull also based on the selection made in the Preferences section. Like the **Delete Existing Data** option, **Perform Sync on Job Start** will only sync once when the job is newly added and started. If you wish to perform this task later, you will have to remove any existing Almanac Sync job and re-add it with this option selected. Please see **Figure 12-1**.

**Figure 12-1**

The screenshot shows the 'Add Jobs to Queue' dialog box. At the top, a text box contains instructions: 'Specify additions to the job queue here. To add a job, select the job type, specify appropriate options, and select the "Add to Queue" button. All are added in a paused state and must be started from the main screen.' Below this, on the left, is a 'Job Type:' dropdown menu with 'Almanac Sync' selected. Underneath are two checkboxes: 'Delete Existing Data' and 'Perform Sync on Job Start', both of which are unchecked. At the bottom left is an 'Add to Job Queue' button. On the right side of the dialog is a table with four columns: 'Queue ID', 'Job Type', 'File Location', and 'File'. The table is currently empty, displaying the message 'No content in table'. A 'Close' button is located at the bottom center of the dialog.

### Job Type: Import

The purpose of the Import job type is to read in observational data from a WeatherLink data file and input that data into our database. Once you have selected **Import** for the job type, you will be able to select the file to import. If you do not see any files listed, check your **Data File** setting in Preferences. The last option to set is **Overwrite if data already exists**. This is useful if you have a previous import job that failed, and you want to replace that data. If selected, wx2sql will analyze the data file for the date/time range of the observational data, remove that data from the database, then begin the import process from the beginning. Please see **Figure 12-2**

**Figure 12-2**

This screenshot shows the 'Add Jobs to Queue' dialog box with 'Import' selected in the 'Job Type:' dropdown menu. The 'Select File:' section now includes a 'Select File' button. The rest of the interface, including the instructions at the top, the 'Delete Existing Data' and 'Perform Sync on Job Start' checkboxes, the empty table with 'No content in table', and the 'Add to Job Queue' and 'Close' buttons, remains the same as in Figure 12-1.

### Job Type: Monitor

The purpose of the Monitor job type is to monitor the active WeatherLink data file for changes, parse the file, and input new data into the database. After selecting the **Monitor** as the job type, there are no other options to configure. Ensure the **Directory to Monitor** is correct, that it points to the 'live' WeatherLink directory that contains the \*.wlk files. If that is correct, select **Add to Job Queue**. Please see **Figure 13-1**.

## Job Types and Queues *(continued)*

**Figure 13-1.**

Specify additions to the job queue here. To add a job, select the job type, specify appropriate options, and select the "Add to Queue" button. All are added in a paused state and must be started from the main screen.

Job Type: Monitor

Directory to Monitor: C:\WeatherLink\VP21

Add to Job Queue

Queue ID	Job Type	File Location	File
No content in table			

Close

## Understanding the Database

### Observation Data

With few exceptions, observation data is stored in the database in the same format as it was read in from the WeatherLink data files. As such, the data cannot be used exactly as it is extracted from the database. Below is a translation table that can be used to format the data in a way that makes it usable.

Column Name	Description	Necessary Formatting
obsn_ep	Time of observation in epoch time	Convert epoch to calendar date
out_tmp	Outside Temperature	Divide value by 10
in_tmp	Inside Temperature	Divide value by 10
bmeter	Barometric Pressure	Divide value by 1000
out_hum	Outside Humidity	Divide value by 10
in_hum	Inside Humidity	Divide Value by 10
wnd_spd	Wind Speed	Divide Value by 10
wnd_dir	Wind Direction	Stored as a 16 point compass value 0 = north, 4 = east, etc. 255 = no value
rain_clicks	Number of clicks from rain sensor	If your rain type = 4096, then for each click, you will have received .01" of precip. For example, if you have recorded 30 clicks (30*.01), you will have received .3" of precipitation
rain_type	Type of rain collector. Used in calculating rain measurements	4096 = each click denotes .01" of rain

Simple SQL queries can be written to extract data to be used in your application. An important thing to keep in mind is that, for all observation data, times are stored in Epoch time. Epoch time is the number of seconds that have elapsed since 00:00:00 UTC, Thursday, January 1<sup>st</sup>, 1970. Many languages have built in functions that will convert epoch time to a human readable time stamp. Below are a couple of links that will provide you with more information.

### **Wikipedia – Unix Time**

[https://en.wikipedia.org/wiki/Unix\\_time](https://en.wikipedia.org/wiki/Unix_time)

### **Web app to convert epoch time to human readable and back.**

<http://www.epochconverter.com/>

### Almanac Data

The almanac tables are more normalized than the observational data, thus pulling data from these tables is a little more complicated. Below are sample SQL queries to pull information from the DB for January 8<sup>th</sup>, 2017

#### **Current Phase**

```
SELECT lk_moon_phase.phase FROM almanac LEFT JOIN lk_moon_phase ON  
almanac.curphase=lk_moon_phase.ID WHERE month='1' AND day='8' AND year='2017';
```

#### **Location (including City, State, and County)**

```
SELECT location.city, location.county, location.state FROM almanac LEFT JOIN location ON  
almanac.location=location.ID WHERE month='1' AND day='8' AND year='2017';
```

#### **Sunrise**

```
SELECT sun.time FROM almanac LEFT JOIN sun ON almanac.sundata=sun.date WHERE month='1' AND day='8'  
AND year='2017' AND sun.phen='R';
```

More queries will be added here for examples at a later date.

## **Error Correction**

When it comes to recording individual observations, invalid data is a distinct possibility. The format docs specify invalid data values such as 32768 or 255, depending on the data type. wx2sql recognizes other values as invalid, such as a negative humidity value, or a humidity value that exceeds 100%. However, the invalid data that wx2sql is most concerned with are invalid 'Packed Time' and 'Archive Interval' values. Packed time is defined as the "minutes past midnight of the end of the archive period. Archive Interval is the number of minutes contained in the archive record.

Keeping observations in the correct chronological order is very important to ensuring data integrity. If an observation record contains an incorrect time, wx2sql will do its best to calculate the correct value and perform many tests to ensure the value is correct. Below is an example from actual data illustrating invalid Packed Time and Archive Interval Values.

Packed Time	Archive Interval	Temp	Hi Temp
1133	1	386	386
1134	1	386	386
61	0	34	97
117	0	105	108
77	0	105	99

When browsing this data from within WeatherLink, the invalid times are not displayed. Wx2sql, when browsing the data file, will read/display the data as-is. It is important for wx2sql to calculate the appropriate Packed Time so the file structure remains consistent and the observation is correctly removed when deleting observations for a specified month. A Packed Time is determined to be invalid if it matches any of the following criteria:

- Packed Time value is less than 1 or greater than 1440
- Archive Interval that are not set to the following values: 1, 5, 10, 15, 30, 60, or 120. These are the only values allowed within WeatherLink. Any other value is invalid
- The packed time currently being processed has not already been processed for the day being processed. In the example 1133 and 1134 are valid times, 61, would be invalid as that value would have already been processed.
- wx2sql determines the Archive Interval being used, and uses that value to calculate what the Packed Time value should be. If the calculated value does not match the current value, it is marked as invalid. There are instances where wx2sql might not be able to determine the Archive Interval. If this problem occurs, you can override the Archive Interval calculation by hard coding the Archive Interval in the Preferences screen.

Please refer to the section on the Preferences screen in this document.

- Any given Packed Time value should be greater than the one that came before it and less than all valid values after it. If this is not true, the value is marked as invalid.

## Error Correction *(continued)*

If the Packed Time value is determined to be invalid, wx2sql will calculate what the value should be and will use that value. Due to that value being invalid, all other data in that record cannot be trusted and will be marked as null.

When repairing invalid Packed Time values, wx2sql will attempt to calculate the correct value by using the initial Packed Time value and Archive Interval and count forward, then use the last Packed Time value and Archive interval to count back. If the calculated values match it will be used. If the calculated values do not match, the process will fail with an Error Code. Please refer to the section on Error Codes in this document .

## Error Codes

Beginning with 0.1.0 Beta 3, errors occurring in wx2sql will display an Error Code. This will help keep the Log Viewer cleaner by not displaying error descriptions every time an error occurs. Please refer to Table X for Error Code definitions. In 0.1.0 Beta 3, this functionality is new and has not been implemented throughout the entire application. Further implementation will occur in future releases.

Error Code	Description
101	Packed Time validation failed, the value has fallen out of the expected range of 1 – 1440. The error that is logged will contain the Day, Record, and Value that was found.
102	Archive Interval validation failed, the value found was not valid. Valid values for Archive Interval are: 1, 5, 15, 30, 60, and 120. The error that is logged will contain the Day, Record, and Value that was found.
103	Duplicate detection validation failed, a duplicate of the value we are processing has already been processed. The error that is logged will contain the Day, Record, and Value that was found.
104	Packed Time validation failed, the value being processed is less than previous valid values. All previous Packed Times should be less than the current value.
105	Packed Time validation failed, the value being processed is greater than future valid values. All future Packed Times should be greater than the current value.
200	Repair Packed Time Failed, validation checks on the repaired Packed Time returned an unexpected value.

## 0.1.4 Changelog

- **About Screen – Check For Updates.** The feature used to check for updates was broken. A complete recode of this section was done to correct the problem.
- **Import Job – Wind Value.** Versions of wx2sql prior to 0.1.4 contained a bug that affected the maximum wind speed and hi wind speed values that could be inserted into the database. The bug caused values that were above 30 mph to be inserted into the database as a null value. To correct the issue, data files containing wind speed values above 30mph will need to be reprocessed.
- **Database Configuration.** Corrected UI bugs in the Database configuration area. Also corrected a bug that affected users that were trying to use DB prefixes.
- **Database UI.** Corrected an issue that under some circumstances, prevented DB tables from being created in the production database

## 0.1.3 Changelog

- **Database Schema Change.** In preparation for support of multiple Davis Weather Stations, a new column was added to the database in order to store a stationID.
- **Database UI.** Database upgrades via a SQL script has been phased out. A new option was added to the Database Screen that allows quick analysis and upgrade of the database schema.
- **Database Configuration.** Previous to version 0.1.3, both the test and production database names were hardcoded within the application. A user inquiry pointed out that some hosting providers require the use of a prefix to the database name. A configuration option was added to support the use of a prefix to the database names.
- **Observational Data Export.** Data Export functionality was added to the WLK File Utilities section. This allows the user to export data stored in a WeatherLink observation file to a CSV file. Data can be exported either in raw or converted format. Observational times can be displayed in the native 'packed' format, a complete timestamp adjusted to the system's default time zone, or a complete timestamp adjusted to a user specified UTC offset.
- **Monitor Job Fix.** A fix was added to address the issue of the monitor job failing when the date is rolled to the next month.
- Various bug fixes and minor enhancements.

## 0.1.2 Changelog

- **Enhanced Logging System.** Format of log viewer was enhanced with improved layout and color coding of log severity. New configuration options were made available that allow you to limit the number of log entries that are retained.
- **UI Overhaul.** The default theme was changed to be not as dark.
  - A tree view was added to the main window allowing you to view a job queue overview, perform database maintenance tasks, and view details of the job queue.
  - Configuring credentials for the production and test databases was made a lot easier. You can now select which database you wish to be active while specifying credentials separately for each database.
- **Logging.** The log viewer was redesigned with a more rigid layout to ensure items are laid out in a more organized design. Options were added to limit the number of log entries that are stored. When the number of entries exceed this limit, older items are removed.
- **Under the Hood.** Several changes in the way wx2sql handles date creation and conversions. Moved away from older methods of doing this to Java 8 methods.
- **Under the Hood.** Minor changes were made to the Import behaves. Trying to smooth out some odd behavior we've observed with this process.
- **Under the Hood.** Upgraded MYSQL JDBC Driver to 5.1.44

## 0.1.1 Changelog

- Added feature that allows the user to encrypt the database password that is stored within the configuration.
- Added WLK Utilities section. The first tool made available in this release will allow you to perform a health check on your WeatherLink data files to look for potential problems.
- Fixed bug that required the monitor process be restarted in order to begin monitoring a new month's data file.
- Numerous bug fixes.



## 0.1.0 Final Changelog

- Added feature that allows wx2sql to fetch moon rise/set, sun rise/set, and moon phase data from the United States Naval Observatory.
- Database Schema version was incremented. This was to accommodate the new Almanac feature. To upgrade the database to the latest Schema, download and run the database script.
- Fixed an issue that caused the wx2sql Monitor process to go into the Abort Status upon a data connection failure, resulting in the process to stop. Now the process will back out of the operation and continue to run.
- Enhanced the shutdown process by adding a UI component that provides the user information on the shutdown process. Also modified the various processes to behave more orderly upon receiving a shutdown request. As a result, the application no longer appears to freeze while shutting down.
- Numerous UI enhancements, including a redesign of the Preferences Menu.
- Bug fixes.

## 0.1.0 Beta 3 Changelog

- Fixed an issue that prevented data from being input into the database when encountering unexpected data values. This required changes to the DB Schema as well.
- Database Schema version was incremented. **Due to changes made, existing databases will need to be dumped and recreated with the updated database creation script.** Existing observation data will be lost and the data will need to be re-imported.
- Redesigned the "Validate Database" process.
- Made efficiency improvements to the "Import" process. This sped up the file import process.
- Added functionality to use known good observation times to accurately correct and replace invalid observation times. Invalid times can be a single occurrence or many occurring in succession.
- Began implementation of Error Codes. This will lead to better Error logging. Error Codes will be further implemented in later versions.
- Changed the database connection routine so that it is validating the DB Schema.
- Implemented a totally new error correction model. Wx2sql is now able to appropriately handle and process invalid data.
- As part of the revamped error correction model, logic for calculating the Archive Interval has been added.
- Added the ability to override the Archive Interval for error correction. The user is now able to set the override and specify a value. This has resulted in new Configuration options under Preferences.
- Added the ability to specify a production or test database. This will allow you to conduct testing in a test database before processing in the production database.

## 0.1.0 Beta 2 Changelog

- Added the ability to Export and Import your configuration file.
- Fixed an issue with the logging subsystem that prevented log events from being displayed when the counter reached ~15,000 entries.
- Switched installer program to something a little less primitive. This results in a smaller setup program, but you are now required to have Java installed on your machine.

## Planned Improvements

- As of 0.1.2, wx2sql only supports one weather station. Support for up to 5 weather stations is in the works. Version 0.1.5 will continue laying the groundwork for this support through various backend and database enhancements.
- The workflow for browsing data files is generally useless when trying to view a specific file. A user is required to scroll through files instead of directly specifying the desired file. This enhancement is planned for 0.1.7.
- Importing multiple data files is currently a slow and tedious process. This process will begin to be improved in 0.1.7.
- Implement secure communication between wx2sql and the database. Targeted version for this feature is not yet specified.
- Implement the ability to edit data stored in the database. Targeted version for this feature is not yet specified.
- Implement the ability to edit .wtk files. Targeted version for this feature is not yet specified.
- Implement the ability to export data from the database to a valid .wtk file. Targeted version for this feature is not yet specified.
- Implement the ability to repair a corrupted .wtk datafile. Targeted version for this feature is not yet specified.

For comments or questions, please contact us at [webmaster@hoosierweather.com](mailto:webmaster@hoosierweather.com). For the latest wx2sql news and updates, follow us on Twitter @HoosierWeather.