

How to Export Data from a dataTaker to a Sequel Database

SQL Database Allows Users to Parse Data Using Filters

At [CAS DataLoggers](#) many of our callers want an automated method to transfer the data from a data logger into a Sequel database so they can easily track and parse the data using filters, create reports and charts, etc. SQL is much more functional than many other databases such as MS Excel, especially when it comes to viewing a large volume of files, say over a period of two years.

[DataTaker data loggers](#) can accomplish this using **FTP** (File Transfer Protocol) to push all the data they record to an FTP server. We've created a program that, whenever it sees a new file, tells the dataTaker to place it directly into MS Sequel. This way, all files are in a convenient database where users can quickly search for data by time, temperature, etc. In our latest **Tech Note**, we detail how to export your collected data from a dataTaker into a Sequel database.

Data Logging Schedules:

In the dataTaker logger we have two logging schedules.

- The first schedule tells the data logger to read 4 separate thermocouple sensors once every 30 seconds and to save the data.
- The second schedule runs once every 6 hours and grabs any and all new data that have been saved from the first schedule, pushing it to an FTP server.

This functionality is already built into the logger via dEX software.

In summary, our data logging schedules are commanding the dataTaker to record 4 temperatures every 30 seconds and 6 hours. When the file is created, its filename format is .csv. In our example we've used FileZilla, a free FTP server available online.

Note: We're saving the data in CSV format, i.e. a text format, so be aware that in the image below, we've set the data format in the data logger so that it's sending the time as integer seconds. This is because databases don't easily deal with fractional seconds.

Data Transfer via FTP:

FTP bundles the collected data from the data logger and sends it to the server. After the server responds, FTP logs on and then sends the data in chunks. FTP also has features within the standard to verify that data is transferred, including filename extensions which we'll cover below. FTP is also popular for its built-in security and redundancy features.

FileZilla grabs all the data and stuffs it into a local directory, uploading it to an upload folder. In our application this is all done by Ethernet.

There are two pieces of software running here:

1. Windows Batch File: When it runs, this application checks to see if there's a new .CSV file in the Upload Folder. When it sees one, it creates a new file called Import Data. Windows Batch File then makes a copy of the file and tells Sequel to load it into the Sequel database. When done, it names the file based

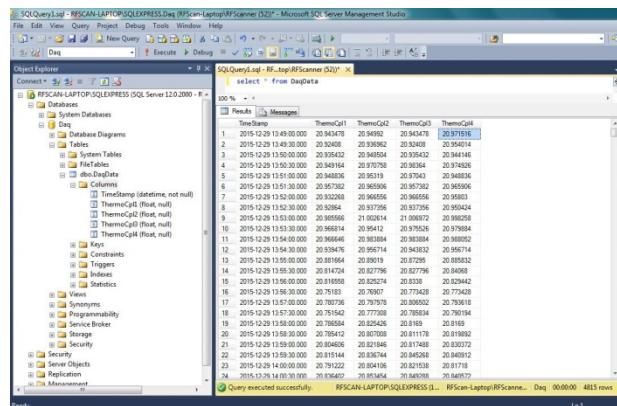
on whether or not the file was successfully uploaded (see below for detail on the .bak and .bad file extensions).

2. Windows Task Scheduler: See below

Sequel Import:

We created a database file named 'Daq' based on a Sequel command called BulkInsert. When it receives the BulkInsert command, Windows Batch File grabs data from the CSV file and loads it into the SQL database, using a format file to create the 'Daq' file. It does this using an SQL format command, in this case 'SQLCHAR.'

Inside Daq, we created a table called 'DaqData'. This table has 5 items in it to track the time stamp along with the 4 temperature values from the thermocouples. See the image below.



This dataTaker capability makes it easy for users to filter their data and present it however they want. It's especially useful for parsing data from one or more years' worth of readings. For instance, in SQL you can run Queries to parse the data, or charting, or 3rd-party reporting, etc.

Automated Setup with Windows Task Scheduler:

While users can execute the 'ConversionLoop' Batch File manually, it's more convenient to automate this process using the [Windows Task Scheduler](#). We do this by using the Task Name 'Upload to SOL.'

For example, in our setup, the Task Scheduler is programmed to run every 10 minutes to see if a new file is there. If so it runs ConversionLoop and imports the file(s) to SQL.

The last function that ConversionLoop performs occurs once it's successfully opened the CSV file. It either:

- Renames the file with the extension .bak (indicating that the data **is** backed up).
 - Or renames it with the extension .bad (indicating that the data **not** backed up).

For more information on how to export data from a dataTaker to a sequel database, contact a CAS DataLoggers Technical Specialist at (800) 956-4437 or visit our website at www.DataLoggerInc.com.