External Datalogging for Hydraulics Platform – Pulling and Parsing

Guide to accessing and using the Force Logic MDL Engineering Dashboard

07/09/2025 – Rev1

# Pulling Datalog Entries

*If you already know how to pull, and want to learn how to use the parser, skip to* [*Parsing Datalog Entries*](#_Parsing_Datalog_Entries)

*If you already have parsed CSV Datalog entries, and want to learn how to use the MDL Engineering Dashboard to View the Data, skip to* [*Viewing Datalog Entries*](#_Using_the_MDL)

## Materials

1. Tool or PCBA
2. 2800 Adapter
3. Battery

## Connect OpenLink Adapter

Connect the OpenLink adapter between the tool and the battery, and then tool to the open link adapter. Then connect the openlink adapter to your computer via a USB-C Cable



Electronics assembly connected to 2800 adapter

## Connect to Tool via QuickLink

Navigate to <https://quicklink.milwaukeetool.com/> on your favorite browser

A computer screen shot of a computer

Description automatically generated

Quicklink Web

After logging in, confirm you are on in **engineer mode**. By default, it will be in technician mode. If your screen looks like the image below, you are in technician mode.

A screenshot of a computer

Description automatically generated

Quicklink Technician Mode

To switch to Engineer mode, click the drop down next to your profile in the top right corner:

A screenshot of a computer

Description automatically generated

Profile dropdown to change settings

Click **Change Mode** and a popup will appear of the different modes. Click Engineer to enter engineering mode:

A screenshot of a computer

Description automatically generated

Mode selection popup

Once you’ve selected engineer mode, quicklink should look like this:

A screenshot of a computer

Description automatically generated

Quicklink Engineer Mode

To connect to the tool, select “Connect via Adapter”. If this is your first time using this adapter, a popup in the search bar will ask you to pick a serial port

A screenshot of a computer

Description automatically generated

Connect via adapter button

A screenshot of a computer

Description automatically generated

Serial port selection

Choose one of the two ports. If you get a red banner that says, “Failed to Connect”, select “Connect Via Adapter” again and choose the other one. If you continue to have issues, make sure the tool is awake and the adapter is connected to your computer.

A screenshot of a computer

Description automatically generated

Quicklink failed to connect banner. Clicking connect via adapter usually resolves this.

Once the tool is connected, you will be able to see the name, MPBID, memory map version, as well as a log of all the command to and from the tool.

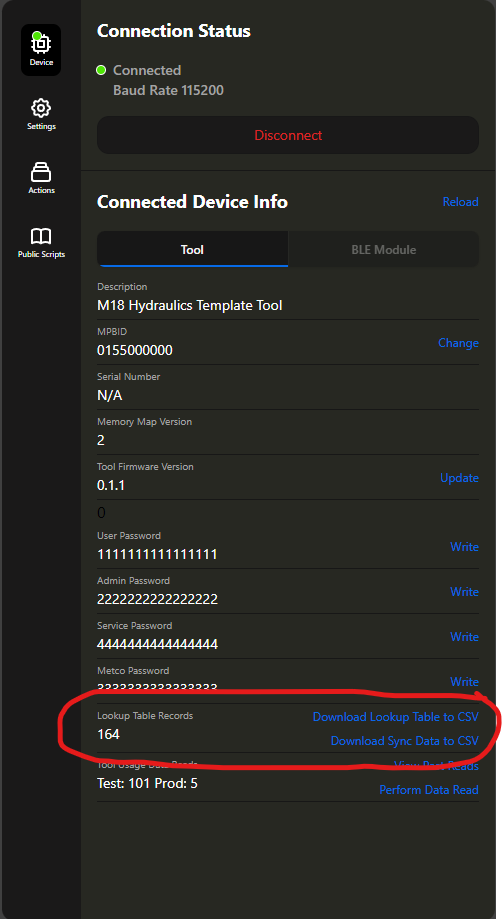
A screenshot of a computer

Description automatically generated

Quicklink Device screen

## Pulling Datalogger Data

Under tool the tool information, you should see **Lookup Table Records**. This number shows how many logs have been stored to the external memory’s lookup table.



Lookup Table Records in Quicklink

If you are trying to pull data and you see 0 records or no Lookup Table Record field, you have an issue and will not be able to pull data.

There are two types of data you can pull, the Lookup Table (or Sequential Data) and Sync Data (or Realtime Data).

### Lookup Table

The lookup table includes a snapshot of data for each cycle. Each lookup table entry corresponds to one cycle. These entries include a variety of things, including peak pressure, battery voltage, temperature, as well as the pointer to the sync data for this cycle.

The current lookup data structure in the template tool will be below, but make sure to check specific tools for any changes.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sync Ptr | Time Stamp | Pass/Fail | Final Pressure | Total Cycles | Cycle Energy | Cycle Distance | Cycle Temperature | Cycle Pack Voltage |
| Battery Impedance | Max Dp | Avg Dp @3k PSI | Avg Dp @8k PSI | Contact Distance | Ramp Rate | Energy In | Time Below | Checksum |

### Sync Data

Sync data tracks certain variables over the course of the cycle. Typically, entries are written every 64ms (unless changed by the EE). Most commonly the variables written include pressure and current, but other variables could be tracked if desired.

The current sync data structure in the template tool will be below, but make sure to check specific tools for any changes

|  |  |  |
| --- | --- | --- |
| Pressure | Distance | Current |

### Pulling the Data

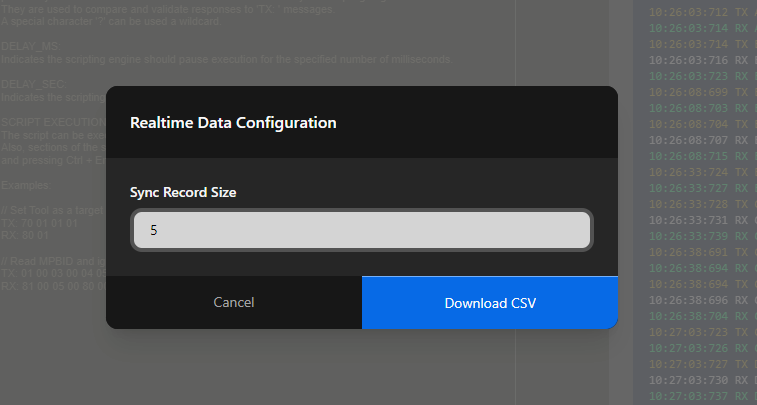
Quicklink provides easy ways to pull both the lookup table and sync data

* To pull the **Lookup Table**, click “Download Lookup Table to CSV”

This will download a csv with the name format: <MPBID>\_raw\_lookup\_<timestamp>.csv

* To pull the **Sync Data**, click “Download Sync Data to CSV”

Enter the size of the sync table **in bytes.** The template tool uses 5 bytes, if your specific tool has changes, confirm the sync size



This will download a csv with the name format: <MPBID>\_raw\_realtime\_<timestamp>.csv

NOTE: Sync data pulls take significantly longer than the lookup table, especially with many cycles. **Make sure the tool stays powered, and the computer stays on for the entire pull.** This may take **hours** if you have thousands of cycles.

If you have downloaded both the sync data and lookup table, your files may look something like this:

A screenshot of a computer

Description automatically generated

Note: if you intend to use the python parser in the next section, it is recommended not to change the name of the files until after converting. If you wish to change the name, you **must leave the raw\_realtime and raw\_lookup\_table parts** so that the program knows what type of file each one is.

# Parsing Datalog Entries

Data that comes off quicklink for non-production tools will just display the raw bytes in hexadecimal. This data needs to be converted to its intended values to be usable.

In this next section I will walk you through how to use a python script to decode these files easily. If you have a different way you want to parse the data, feel free to use that instead.

## Materials:

1. Files to parse from QuickLink
   1. Can be any amount of sync data and/or lookup table records
2. Python 3.11 installed
   1. Can be downloaded in software center
3. Your favorite python IDE
   1. VSCode, PyCharm, Jypiter notebook, etc.
   2. I will not walk you through how to setup these IDEs. Consult online walkthroughs or Milwaukee Tool documentation

## Setup Folder Structure

Navigate to [Hydraulics Datalogging](https://milwaukeetool.sharepoint.com/:f:/t/ee/EpvFIpxUfPpBlt8RDXwHnpoBeIO8BAYPmBXnTxm2w0smTw?e=XMe8yd) folder in SharePoint, and download a copy of the python file hyd\_ext\_datalog\_parse.py and the json that corresponds to your **external** memory map

* V2 matches the memory map map described above
* V3 is a developmental version with added battery variables – capacity, temperature, and voltage
* If you have a custom memory map, download one of the two and modify it as needed

A screenshot of a computer

Description automatically generated

Sharepoint location of python script

Create a new folder for this file in whatever location you want. Place the python file and memory map in this location

A screenshot of a computer

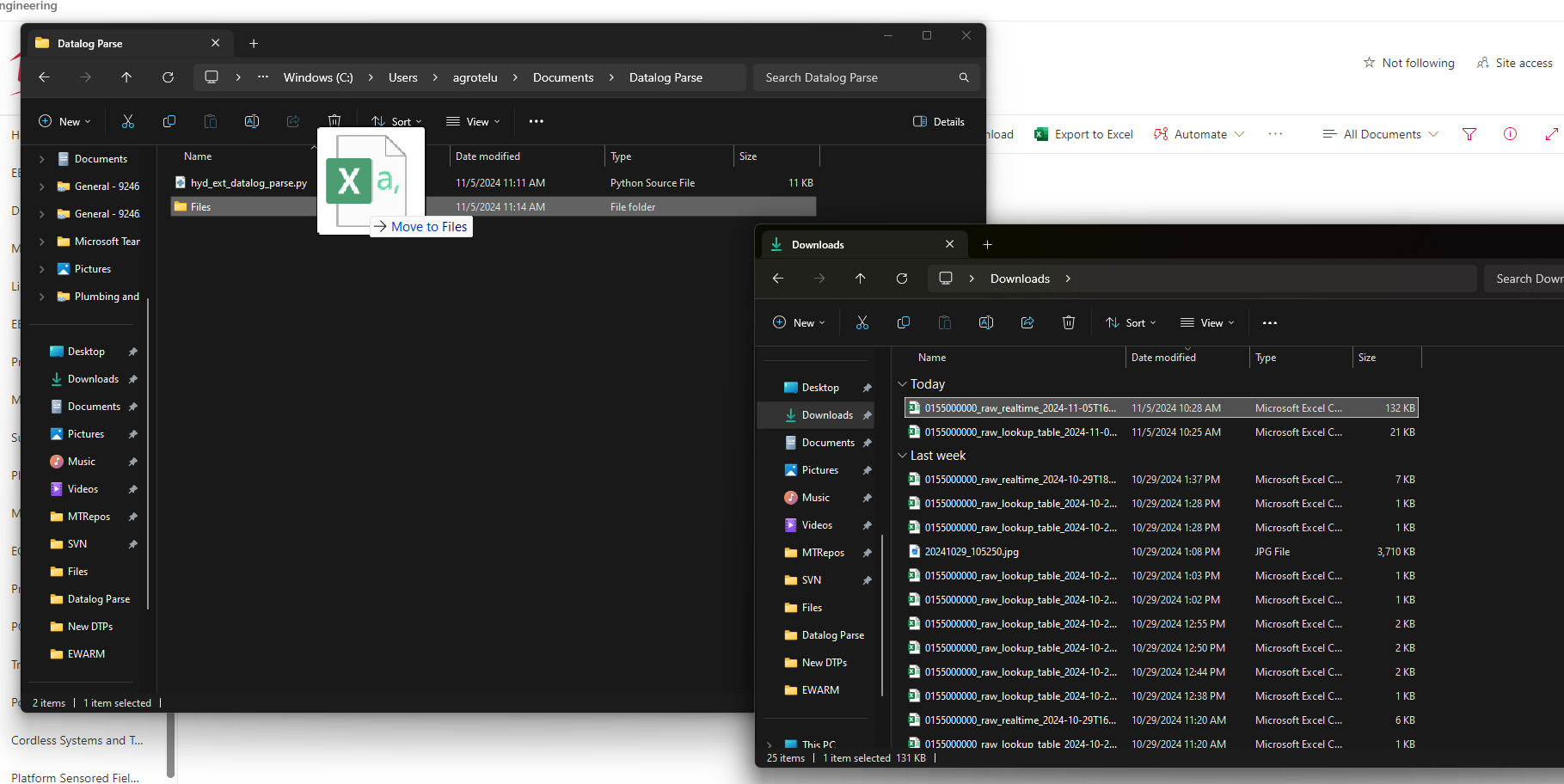
Description automatically generated

Once you have created this folder, create an empty folder in this folder called **“Files”**

A screenshot of a computer

Description automatically generated

In the “Files” folder, you should put the files you want to be converted. **It is recommended to keep the names that QuickLink generated until after the files have been converted.** You can move any number of sync or lookup table records.

****

## Launch IDE

After moving the files you want, we need to open the file in the IDE of your choice. Note: the next steps will be assuming VSCode, but other python IDEs should be similar.

Launch VSCode, and click **File -> Open Folder**

A screenshot of a computer

Description automatically generated

Navigate to the location of the folder you created for the python file. Ensure that the target folder is the folder holding both the python files and the Files folder. **Do not open the Files folder.** You will not see the python file when navigating because it will only show folders.

A screenshot of a computer

Description automatically generated

If opened correctly, you will see the >Files and the python file in the explorer of VSCode. You can expand the Files by clicking the arrow to insure your datalog files are there.

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Open the python file by clicking on it. The program defaults to using mem\_map\_v2. If you are using v3 or your own, change the **memory\_map\_file** name to match your memory map json.

A screenshot of a computer program

Description automatically generated

You should now be ready to run the program.

## Run Program

To run the program, click the play button in the top right corner of VSCode

*You may be prompted to select a python interpreter before running. Select the version of python you have installed.*

## A screenshot of a computer Description automatically generated

If the program runs successfully, you will be prompted in the terminal to select a conversion mode.

A screenshot of a computer program

Description automatically generated

If the program does not run, check a few things

* Make sure the correct python interpreter is selected. This file was created for version 3.11.3 and is not guaranteed to work with python 2
* Make sure all extensions are installed. If you have errors related to the extensions, you may need to install them by typing these commands in the terminal:
  + For os – pip install os
  + For tqdm – pip install tqdm
* Make sure you have opened the folder with the python file and the “Files” folder in itf. The program will be unable to file your files if you are in a different folder

There are four options for conversion:

* All – Converts all cycles in the file
* First 100 – Converts the first 100 cycles in the file. If there are less than 100 cycles, all will convert
* Last 100 – Converts the last 100 cycles in the file. If there are less than 100 cycles, none will convert
* Custom Cycles – You will be prompted for the first and last cycle, and your range will be converted. If the first and last cycle are the same, you will just get that 1 cycle.

You should enter the one letter character corresponding to the options given. The character can be uppercase or lowercase.

A screenshot of a computer program

Description automatically generated

Note: The same conversion scheme will apply to all files found in the “Files” folder. If you want different schemes for different files, you should run the program multiple times with different files in the folder.

Once you enter your choice, all the files will be converted. You will see a message for each file the program finds, as well as a progress bar. Large sync data files may take a minute or two to see the progress bar pop up.

A screenshot of a computer

Description automatically generated

Once the conversion is complete, you should see one extra file for each file with “raw” changed to “conv”.

Lookup table files will be – <MPBID>\_conv\_lookup\_table\_<timestamp>.csv

Sync data files will be – <MPBID>\_conv\_realtime\_<timestamp>.csv

## Use Data

You can now open the converted csv files in your favorite csv editor, like excel.

A screenshot of a computer screen

Description automatically generated

# Using the MDL Engineering Dashboard – by Derek Achen

Once you have parsable CSV files as a product of the steps above of either the Lookup Table Data (or Sequential Data) and the Sync Data (or Realtime Data), you will find that interpreting the data as an Engineer is rather difficult. That is where the MDL Engineering Dashboard comes in, allowing you to easily locate trends in your data

In this next section I will walk you through how to install and use the MDL Engineering Dashboard to quickly upload, process, and view the Sequential and Realtime Data.

## Materials:

1. Properly parsed CSV files (products of the steps above in [*Parsing Datalog Entries*](#_Parsing_Datalog_Entries) ) to upload and view.

a. Can be any amount of sync data (realtime) and/or lookup table (sequential) CSV files formatted similarly to the results above. Ensure the ‘ApplicationId’ column exists in both these files.

1. Python 3.11 installed
   1. Can be downloaded in software center
2. Your favorite python IDE
   1. VSCode, PyCharm, Jypiter notebook, etc. any python IDE.
   2. I will not walk you through how to setup these IDEs. Consult online walkthroughs or Milwaukee Tool documentation

## Setup Folder Structure

Navigate to [Hydraulics Datalogging](https://milwaukeetool.sharepoint.com/:f:/t/ee/EpvFIpxUfPpBlt8RDXwHnpoBeIO8BAYPmBXnTxm2w0smTw?e=XMe8yd) folder in SharePoint, and download a copy of the folder titled “Achen-MTProject2025” containing the following Python source files and Excel file

main.py

ui.py

data\_loader.py

visualizer.py

ADL Variable Enums.xslx

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a web page

AI-generated content may be incorrect.Download and place this folder wherever you’d like in your file system. If the folder downloads as a zipped folder, please right click on the zipped folder in your file system and click “Extract All”. This will unzip the files and make them available for your use.

Please follow the instructions in the [Launch IDE section](#_Launch_IDE) to open the “Achen-MTProject2025” in your favorite Python IDE

Once in your IDE, open the main.py file

A screenshot of a computer screen

AI-generated content may be incorrect.

Following the instructions to [launch the program](#_Run_Program), launch the UI, plug in your Sequential or Realtime files (or both) and view the data away. Please check the console for any important error messages. Most should be displayed by the programs UI, however.

For a demonstration of the tools capabilities, please navigate to the presentation video in the [Hydraulics Datalogging](https://milwaukeetool.sharepoint.com/teams/ee/PE/Forms/AllItems.aspx?id=%2Fteams%2Fee%2FPE%2FNew%20Product%20Development%2FHydraulics%20Datalogging&viewid=3926cd9f%2Da8c8%2D41e1%2Db7e8%2D1abd704ce9dd&p=true&ga=1) folder .

A screenshot of a computer

AI-generated content may be incorrect.

You may need to run ‘pip install openpyxl’ in the terminal if you are getting errors. Otherwise, any other errors to Alex Grotelueschen

Alex.Grotelueschen@milwaukeetool.com