VEET 2.1 Data Processing Guide

Introduction

This guide provides instructions and best practices for using widely available Windows and macOS tools to manipulate sensor data files of the Visual Environment Evaluation Tool (VEET).

Table of Contents

Validating Data Completeness

To evaluate whether the VEET was properly charged and continuously logging data for the established time of the research project, start by setting a baseline: determine the total number of data lines you expect to find in the VEET sensor data file and compare that to the actual number of data lines in your file.

1. **Establish a baseline of the expected lines of data:** Divide the number of seconds in a day by your sensor-logging interval and multiply by the number of active sensors (e.g. 86,400 / 2 seconds x 4 sensors x 14 days = 2,419,200 expected data points).

2. **Compare to the actual number of data lines recorded:** Count the actual number of data lines in your VEET sensor file. Verify that the number of data lines in the VEET sensor file is within 2% of your baseline.
   NOTE: For various clock and saturation event reasons, the total number of sensed events may differ from the expected by up to 2%. This behavior is normal.

If the number of data lines in your sensor data file falls within 2% of your baseline, you can assume the device was charged and logging sensor data for the entire established time and the data is valid for analysis.

TIP: If the number of data lines in your sensor file is not within 2% of your baseline, identify the longest periods of time for which you have continuous data. Then, determine if this amount of time meets the minimum required for your research project.

Using the VEET Validation Script

Our Python script does high-level analysis and processing of VEET Sensor Data files and is available on the Downloads page of https://projectveet.com. The script was tested on MacOS 15.3 and Windows 11, but may work on other MacOS and Windows versions.

Our Python script is accessible to researchers familiar with editing and running Python code. For more information on how to go through this process without the script, navigate to the Counting and Splitting VEET data sections below.

Converting Unix Time

The VEET records time in the Unix time format. To convert from Unix to human-readable time (e.g. from “1719298799” to “6/25/2024 0:00:00”), find the conversion formula online for your specific computational tool, as each tool handles time slightly differently. Epoch Converter provides instructions for many languages/tools.

For Excel, use the following formula:

=((A1 +/- timezone offset) / 86400) + 25569

Then, convert the cell’s number format to a custom date format: m/d/yyyy h:mm:ss

Accounting for Your Time Zone Offset

When converting from Unix time to human-readable time, make sure to account for your time zone offset. The VEET reports your time zone offset in the Information line of the sensor file. Find more information on how to interpret VEET data lines in the VEET 2.1 Data Interpretation Guide.

Example Information Line with Time Zone offset in bold

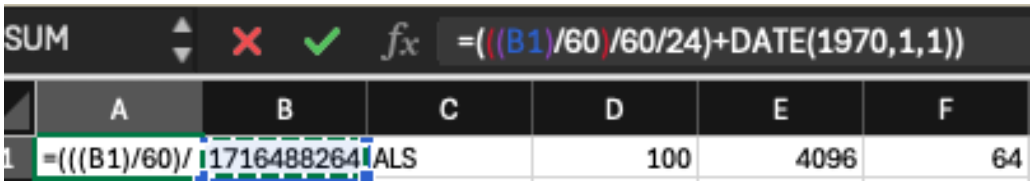
```
1741824121,INF,VEET,24110244,FW2.2.3 Jul 16 2025 11:21:47
,abc123,abc123,-8,2000,2000,2000,2000,L
```

Example: Converting Unix time in Excel
We use the Information line of a split ALS sensor data file to navigate through this example.

- Open the relevant sensor data file on Excel.
- Copy this formula and paste it in an empty cell.

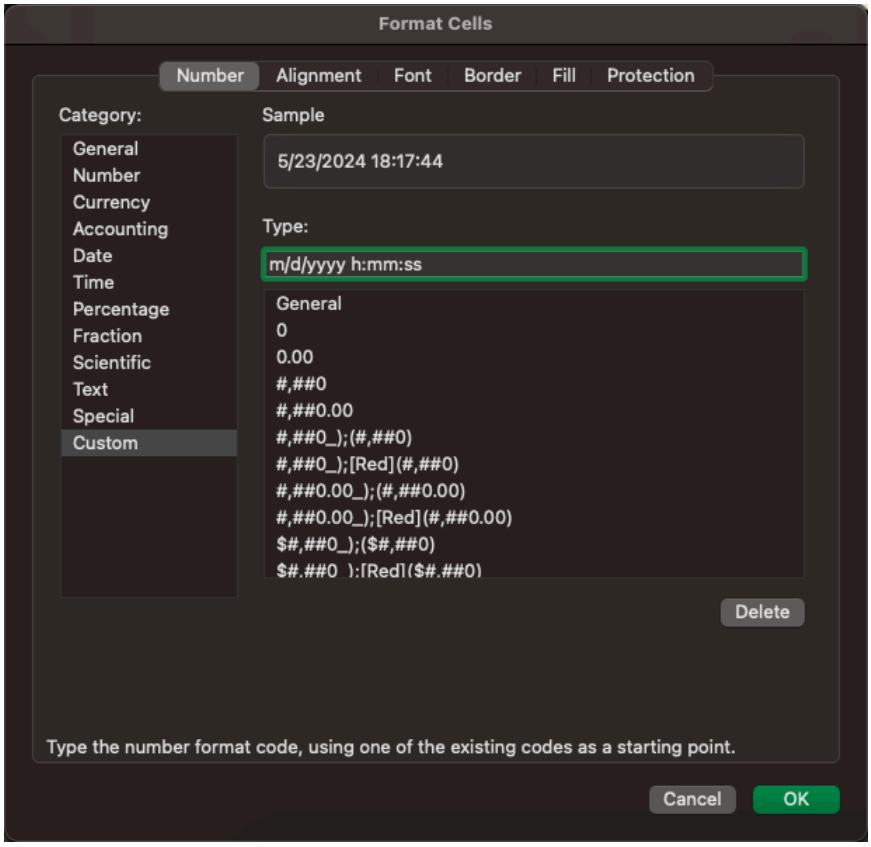
  =(((Cell With Timestamp)/60)/60/24)+DATE(1970,1,1)

- Replace “Cell Name with Timestamp” in the formula with the Unix timestamp you want to convert by selecting its cell (e.g. in the example below, we selected cell B1).



Selecting the Unix Timestamp Cell in the Time Conversion Formula

- Convert the timezone offset to seconds.
  - Our timezone offset was -7. To get that offset in seconds, we multiplied that by 3600 (60 minutes x 60 seconds).
    - -7 x 3600 = -25200
- Subtract the timezone offset from the cell you selected.
  - Our formula looked like this: =(((B1-25200)/60)/60/24)+DATE(1970,1,1))
- Press Enter.
- Right click the cell that contains the formula.
- Select Format Cells > Custom
- Type in the format m/d/yyyy h:mm:ss



Reformatting Cell Time

- Select “OK”
- Navigate to the cell that contains the formula.
- Select this cell’s entire column by clicking on the column letter at the top.
- Type Command + D to apply the formula to the entire column.

Counting VEET Data on Windows

Counting the number of data entries in your VEET sensor file allows you to understand whether the right and left devices logged similar amounts of data and whether the files need to be split or further processed to load into analysis programs (e.g. Excel has a 1,048,576 row limit). These steps were written for and tested on Windows 11 but may work with other Windows versions.

1. Download the latest version of PowerShell.
    - Some of the commands outlined below do not work with the older versions of PowerShell that come with Windows. Make sure that you install the latest one.

2. Run PowerShell 7
    - PowerShell 7 and previous PowerShell versions can run simultaneously (i.e. PowerShell 7 does not upgrade/replace older versions). Make sure you are running the most updated version.

3. Copy the following command and paste it in PowerShell 7:

   ```
   (Get-Content "<<file path>>").Length
   ```

4. Copy the full path of your data file.
   - Find the file in Windows Explorer.
   - Hold down the Shift key, right-click, and select **Copy as Path**

5. Use your keyboard's right and left arrows to navigate to **"<<file path>>"** on the Get-Content command line and replace it with the file path you copied.

   Example:

   ```
   (Get-Content "C:\Right_Sensor_Data.csv").Length
   ```

6. Type Enter anywhere on the command line to get the total number of data lines in your VEET sensor file.

Counting VEET Data on macOS

Counting the number of data entries in your VEET sensor file allows you to understand whether the right and left devices logged similar amounts of data and whether the files need to be split or further processed to load into analysis programs (e.g. Excel has a 1,048,576 row limit). These steps were written for and tested on macOS Sonoma 14.5 but may work on other versions of macOS.

1. Run the Terminal program built into macOS to run Unix commands.
   - Type Command-Spacebar to open Spotlight Search.
   - Type **Terminal** followed by Enter to open a new Terminal window.

2. Copy the following grep -o command and paste it in the Terminal:

   ```
   grep -o "\r" "<<file path>>" | wc -l
   ```

3. Copy the full path of your data file
   - Find the VEET sensor file in Mac Finder
   - Right-click the file, press Option (Alt), and select **Copy "<<file>>" as Pathname**

4. Use your keyboard's right and left arrows to navigate to <<file path>> in the grep -o command and replace it with the file path you copied.
   Example:

```
grep -o "\r" "/Users/alicebob/Desktop/Right_Sensor_Data.csv" | wc
-l
```

NOTE: Make sure to erase the carrots (greater / less than symbols) and to include quotation marks at either end or your file name.

5. Type Enter anywhere on the command line to get the total number of data lines in your VEET sensor file.

Splitting the sensor data on Windows

Splitting a large VEET sensor data file by sensor type yields smaller files that are easier to load and work with in analysis programs like Excel.

1. Run PowerShell 7.
    - For instructions, see steps 1-2 in Counting VEET Data on Windows.

2. Copy the corresponding Select-String command below and paste it in PowerShell 7.

```
Select-String -Pattern "ALS" -Path "<<file path>>" -Raw >
"<<output path for ALS>>"

Select-String -Pattern "TOF" -Path "<<file path>>" -Raw >
"<<output path for TOF>>"

Select-String -Pattern "IMU" -Path "<<file path>>" -Raw >
"<<output path for IMU>>"

Select-String -Pattern "PHO" -Path "<<file path>>" -Raw >
"<<output path for PHO>>"

Select-String -Pattern "INF" -Path "<<file path>>" -Raw >
"<<output path for INF>>"
```

3. Copy the full path of your data file.
    - For instructions, see step 4 in Counting VEET Data on Windows.

4. Use your keyboard's right and left arrows to navigate to **<<file path>>** in the Select-String command and replace it with the file path you copied.

    Example:

```
Select-String -Pattern "ALS" -Path "C:\Right_Sensor_Data.csv"
-Raw > <<output path for ALS>>
```

NOTE: Make sure to erase the carrots (greater / less than symbols) and to include quotation marks at either end or your file name.

5. Replace **<<output path for XXX>>** in the later half of the Select-String command with the name of the new file you want to create.

   Example:

   ```
   Select-String -Pattern "ALS" -Path "C:\Right_Sensor_Data.csv"
   -Raw > "C:\Right_Sensor_Data_ALS.csv"
   ```

   TIP: We recommend using a consistent naming convention that identifies the VEET temple arm and sensor type. For example, if the **<<file path>>** is **Right_Sensor_Data.csv** and you are splitting the file by the ALS sensor, then name the **<<output path for XXX>>** as **Right_Sensor_Data_ALS.csv**.

6. Type Enter anywhere on the command line. Each of the Select-String commands yields a single file based on your output naming convention.

Splitting the sensor data file on macOS

Splitting a large VEET sensor data file by sensor type yields smaller files that are easier to load and work with in analysis programs like Excel.

1. Run the Terminal program on your macOS.
   - For instructions, see Step 1 in Counting VEET Data on macOS.

2. Copy the pertinent sed / grep command below and paste it in the Terminal.

   ```
   sed 's/\r/\n/g' "<<file path>>" | grep "ALS" > "<<output path
   for ALS>>"
   ```

   ```
   sed 's/\r/\n/g' "<<file path>>" | grep "TOF" > "<<output path
   for TOF>>"
   ```

   ```
   sed 's/\r/\n/g' "<<file path>>" | grep "IMU" > "<<output path
   for IMU>>"
   ```

   ```
   sed 's/\r/\n/g' "<<file path>>" | grep "PHO" > "<<output path
   for PHO>>"
   ```

   ```
   sed 's/\r/\n/g' "<<file path>>" | grep "INF" > "<<output path
   for INF>>"
   ```

3. Copy the full path of your data file.
    - For instructions, see step 3 in Counting VEET Data on macOS.

4. Use your keyboard's right and left arrows to navigate to the **<<file path>>** in the sed/grep command and replace it with the file path you copied.

   Example:

   ```
   sed 's/\r/\n/g' "/Users/alicebob/Desktop/Right_Sensor_Data.csv" |
   grep "ALS" > <<outpath for ALS>>
   ```

NOTE: Make sure to erase the carrots (greater / less than symbols) and to include quotation marks at either end or your file name.

5. Replace **<<output path for XXX>>** in the later half of the sed/grep command with the name of the new file you want to create.

   Example:

   ```
   sed 's/\r/\n/g' "/Users/alicebob/Desktop/Right_Sensor_Data.csv" |
   grep "ALS" > "/Users/alicebob/Desktop/Right_Sensor_Data_ALS.csv"
   ```

   NOTE: We recommend using a consistent naming convention that identifies the VEET temple arm and sensor type. For example, if the **<<file path>>** is **Right_Sensor_Data.csv,** and you are splitting the file by the ALS sensor, then name the **<<output path for XXX>>** as **Right_Sensor_Data_ALS.csv**.

6. Type Enter anywhere on the command line. Each of the sep/grep commands yields a single file based on your output naming convention.

Organizing the Data

Once you split the VEET data files into smaller files by sensor type, make the file more manageable to work with by organizing the data. Though Excel is just one of many tools you can use to work with the data, we use it to navigate through the example in this section.

- Open the split sensor data file in Excel or your tool of choice.
    - When we open the example Left_Sensor_Data_ALS.csv file in Excel, it looks like this:

1719298799	ALS	100	4096	4096	4096	354	4374	76956	62	0
1719298801	ALS	100	4096	4096	4096	354	4406	77556	65534	0
1719298803	ALS	100	4096	4096	4096	337	4294	77356	62	0
1719298805	ALS	100	4096	4096	4096	337	4231	77541	62	0
1719298807	ALS	100	4096	4096	4096	320	4151	77448	65534	0

Left Sensor Data ALS Unlabeled

- Use the VEET 2.1 Data Interpretation Guide to identify each value and label each column accordingly.

	D	E	F	G	H	I	J	K	L	M	N
1						Raw Sensor Data					
2	Timestamp	Sensor	Integration Time	uvGain	visGain	irGain	uvValue	visValue	irValue	Flicker	Lux
3	1719298799	ALS	100	4096	4096	4096	354	4374	76956	62	0
4	1719298801	ALS	100	4096	4096	4096	354	4406	77556	65534	0
5	1719298803	ALS	100	4096	4096	4096	337	4294	77356	62	0

Left Sensor Data ALS Labeled