

# Plx 2 mat

v.3 Dec 2019

## How to convert .plx file to m file:

### 1. Pre-requisites:

- Download the latest version of the [plx2mat software](#)
- Copy the files
  - o `convert_plx2mat.m`
  - o `mexPlex.mexw32` (for 32-bit Windows)
  - o `mexPlex.mexw64` (for 64-bit Windows)into your working directory (where you're working on Matlab files).

### 2. Steps:

- Run the command

```
>> convert_plx2mat()
```
- After selecting the file, you will be prompted with two options. Select easy mode:

Please select a mode:

- (1) Easy mode - saves all data, including waveforms and unsorted spikes, to a single file
- (2) Advanced mode - will prompt you to allow selective saving of data, and/or saving to multiple files

Type selection, or press ENTER to accept default (1):

- The resulting .mat file contains the following fields:

| Name                          | Value                                     |
|-------------------------------|---|
| InputFileName                 | '2017-0728-0517_Subject1_Ephys.plx'       |
| SpikeSamplingFreqHz           | 44100                                     |
| Comment                       | 'Simulated device2017-07-28 05:17:31.666' |
| Trodalness                    | 1   |
| PointsPerWaveformPreThreshold | 8   |
| DurationSeconds               | 18.2039                                   |
| RecordingDateTime             | '7/28/2017 5:17:31'                       |
| SpikeData                     | <2x11 cell>                               |
| ContinuousData                | <2x4 cell>                                |
| VideoTrackingData             | <3x4 cell>                                |
| EventData                     | []  |

The fields are as follows:

1. InputFileName The name of the original source PLX file.
2. SpikeSamplingFreqHz The sample rate, in Hz, of *spike* data. Continuous EEG/LFP sample rates will be lower
3. Comment Additional text information provided by some acquisition programs.
4. Trodalness This will either be 1 (single wires), 2 (stereotrodes), or 4 (tetrotodes)
5. PointsPerWaveformPreThreshold For spike recordings, this specifies how many of the points in each spike waveform occur prior to the voltage reaching the threshold for spike detection. This allows part of the recorded waveform to precede the moment when the voltage crosses the detection threshold.
6. DurationSeconds Total duration of the recorded file, in seconds.
7. RecordingDateTime The time and date when recording began.
8. SpikeData Cell array of spike timestamps and waveforms (see details below)
9. ContinuousData Cell array of EEG/LFP continuous data (see details below)
10. VideoTrackingData Cell array of X-Y video tracking data, if available.
11. EventData Cell array of event timestamps.
12. DataGaps (Not always present) Timestamps of JAGA recording interruptions, e.g. from lost WiFi packets.

### 3. SpikeData Cell Array

This array has 11 columns:

| SpikeData <16x11 cell> |           |                       |                   |          |          |          |                      |                    |               |          |          |
|------------------------|-----------|-----------------------|-------------------|----------|----------|----------|----------------------|--------------------|---------------|----------|----------|
|                        | 1         | 2                     | 3                 | 4        | 5        | 6        | 7                    | 8                  | 9             | 10       | 11       |
| 1                      | 'Name'    | 'Timestamps unsorted' | 'Unit 1'          | 'Unit 2' | 'Unit 3' | 'Unit 4' | 'Waveforms unsorted' | 'Unit 1'           | 'Unit 2'      | 'Unit 3' | 'Unit 4' |
| 2                      | 'Spike1'  | <285960x1 double>     | <7751x1 double>   | []       | []       | []       | <285960x25 double>   | <7751x25 double>   | []            | []       | []       |
| 3                      | 'Spike2'  | <426252x1 double>     | <47588x1 double>  | []       | []       | []       | <426252x25 double>   | <47588x25 double>  | []            | []       | []       |
| 4                      | 'Spike3'  | <390379x1 double>     | <114821x1 double> | []       | []       | []       | <390379x25 double>   | <114821x25 double> | []            | []       | []       |
| 5                      | 'Spike4'  | <368630x1 double>     | <32785x1 double>  | 194.6633 | []       | []       | <368630x25 double>   | <32785x25 double>  | <1x25 double> | []       | []       |
| 6                      | 'Spike5'  | <658187x1 double>     | <165574x1 double> | []       | []       | []       | <658187x25 double>   | <165574x25 double> | []            | []       | []       |
| 7                      | 'Spike6'  | []                    | []                | []       | []       | []       | []                   | []                 | []            | []       | []       |
| 8                      | 'Spike7'  | <329436x1 double>     | <75400x1 double>  | []       | []       | []       | <329436x25 double>   | <75400x25 double>  | []            | []       | []       |
| 9                      | 'Spike8'  | <182783x1 double>     | <45639x1 double>  | []       | []       | []       | <182783x25 double>   | <45639x25 double>  | []            | []       | []       |
| 10                     | 'Spike9'  | <199012x1 double>     | <39916x1 double>  | []       | []       | []       | <199012x25 double>   | <39916x25 double>  | []            | []       | []       |
| 11                     | 'Spike10' | <563613x1 double>     | <26300x1 double>  | []       | []       | []       | <563613x25 double>   | <26300x25 double>  | []            | []       | []       |
| 12                     | 'Spike11' | []                    | []                | []       | []       | []       | []                   | []                 | []            | []       | []       |
| 13                     | 'Spike12' | <729423x1 double>     | <100773x1 double> | []       | []       | []       | <729423x25 double>   | <100773x25 double> | []            | []       | []       |
| 14                     | 'Spike13' | []                    | []                | []       | []       | []       | []                   | []                 | []            | []       | []       |
| 15                     | 'Spike14' | <326611x1 double>     | <27643x1 double>  | []       | []       | []       | <326611x25 double>   | <27643x25 double>  | []            | []       | []       |
| 16                     | 'Spike15' | <606376x1 double>     | <85867x1 double>  | []       | []       | []       | <606376x25 double>   | <85867x25 double>  | []            | []       | []       |

The first row is a “header” row. After this, each subsequent row corresponds to one spike channel. Hence, if there are n spike channels, there will be n + 1 rows in the cell array.

The columns in the cell array are as follows:

1. Channel names.
2. Timestamps of unsorted spikes, in ascending order.
3. Timestamps for first sorted unit.
4. Timestamps for second sorted unit.
5. Timestamps for third sorted unit.
6. Timestamps for fourth sorted unit.
- 7-11. Waveforms corresponding to spikes in columns 2-6. Waveforms are also sorted in ascending temporal order, in order to exactly match the timestamps in columns 2-6.

### Data extraction for SpikeData:

- To access "Spike1" unsorted waveforms: Arrow red in the Cell array above

```
>> load ('.\file-name.mat')
```

Make sure to specify the file paths/Directories

```
>> s = SpikeData{2,7};
```

This will populate s with 285960 spike forms with each 25 data points. -> See arrow above

- To access the nth row:

```
>> SpikeData{2,7} (n,:)
```

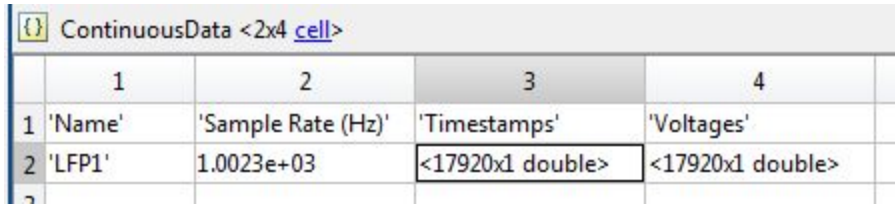
## 4. Continuous Data Cell array: For EEG, LFP or raw data for spikes

To access "ContinuousData" LFP waveforms: Arrow red in the Cell array above

```
>> load ('.\file-name.mat')
```

Make sure to specify the file paths/Directories

This Matlab cell has 4 columns:



|   | 1      | 2                  | 3                | 4                |
|---|--------|--------------------|------------------|------------------|
| 1 | 'Name' | 'Sample Rate (Hz)' | 'Timestamps'     | 'Voltages'       |
| 2 | 'LFP1' | 1.0023e+03         | <17920x1 double> | <17920x1 double> |

The columns are as follows:

1. Channel names
2. Sample rate in Hertz. This rate will generally be lower than the spike sample rate.
3. Timestamps. Timestamps of each datapoint, in ascending order.
4. Voltage values. Voltage values for each datapoint, matching the timestamps in column 3.

The arrays in columns 3 and 4 will always have the same number of elements. Timestamps are always in ascending order. If there are gaps in acquisition, e.g. if packets were lost, there may be corresponding gaps in the timestamps.

The first row contains the names of the columns, so to access 'LFP1' we need to access row 2, and so on. That's why there's an 'n+1' in the access instructions below.

- To access the Timestamps for the nth channel:

```
>> ContinuousData{n+1,3} (1:m,:) % m is the number of data points to load
```

- To access the Voltage level for the nth channel:

```
>> ContinuousData{n+1,4} (1:m,:) % m is the number of data points to load
```

- To grab all the Voltage levels for the nth channel and plot them:

```
>> v = ContinuousData{n+1,4} (:,:); % n is the channel you want  
>> plot(v)
```

## 5. Creating absolute timestamps

The timestamps columns above contain *relative* timestamps measured from the beginning of the capture. Unfortunately, the beginning timestamp is only captured in a text comment, due to limitations of the plx storage format.

To retrieve the absolute timestamp (in local time) of the start of the experiment:

```
>> Comment
Comment =

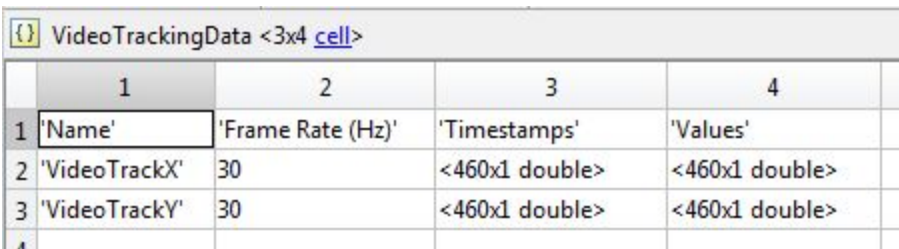
    'JAGA16 wireless device
    2018-01-21 11:24:07.622'
```

The second line shows the timestamp in ISO format. The MATLAB datetime conversion expression to use is:

```
datetime({'2018-01-21 11:24:07.622'}, 'InputFormat', 'yyyy-MM-dd HH:mm:ss.SSS')
```

## 6. Videotracking Data

If video tracking data is available, then this cell array will contain X and Y location data formatted the same way as the continuous data stream. The frame rate will typically be 30 Hz.

A screenshot of a MATLAB variable viewer window titled 'VideoTrackingData <3x4 cell>'. The window displays a table with 4 columns and 4 rows. The columns are labeled '1', '2', '3', and '4'. The rows are labeled '1', '2', '3', and '4'. The first row contains the headers: 'Name', 'Frame Rate (Hz)', 'Timestamps', and 'Values'. The second row shows 'VideoTrackX' with a frame rate of 30, and two columns of '<460x1 double>'. The third row shows 'VideoTrackY' with a frame rate of 30, and two columns of '<460x1 double>'. The fourth row is empty.

|   | 1             | 2                 | 3              | 4              |
|---|---------------|-------------------|----------------|----------------|
| 1 | 'Name'        | 'Frame Rate (Hz)' | 'Timestamps'   | 'Values'       |
| 2 | 'VideoTrackX' | 30                | <460x1 double> | <460x1 double> |
| 3 | 'VideoTrackY' | 30                | <460x1 double> | <460x1 double> |
| 4 |               |                   |                |                |