Tips and tricks for using Lab Streaming Layer (LSL)

Although many people are using Lab Streaming Layer (LSL) in their EEG setups, there are still some potential issues one can encounter when using LSL. With these tips and tricks for using LSL, we want to point out potential solutions and ideas to make your experience using LSL and EEG acquisition as smooth as possible.

> **Disclaimer:** Brain Products is not the developer of LSL. The suggestions and tips provided in this article are based purely on our own experience and we do not guarantee that our recommendations follow the best possible practice. For more information, please also consult additional information, for example this publication from the creators, the main LSL website, the documentation, or here on GitHub.

Tips and tricks for using LSL

LSL offers a convenient solution for synchronizing multimodal data streams with heterogenous sampling rates. It is possible to combine many outlets and inlets without having to worry about a shared hardware clock and the user typically does not have to deal with the internal, complex mechanics.

A vast network, open-source software, APIs for all kinds of programming languages, and the existence of many read-to-use tools have made LSL very popular.

We at Brain Products have supported LSL for many years. All our current amplifiers have dedicated LSL connectors that can be used instead of or in addition to our own data acquisition software, BrainVision Recorder. We also provide a useful and easy-to-use online signal viewer, the LSL Viewer, and recently, we have even upgraded our TriggerBox Plus to work as an out-of-the-box LSL marker source.

> ***Check out our other articles and resources regarding LSL:***
>
> - *The Brain Products GitHub page, hosting LSL connectors for all of our current amplifiers*
> - *A blog post discussing the combination of LSL and hardware triggers*
> - *A blog post walking through the general LSL pipeline*
> - *A press release article explaining how to do hyperscanning with Brain Products equipment and LSL*
> - *The LSL Viewer (introduction and download link)*
> - *Introducing the new TriggerBox Plus with integrated LSL functionality*

- *Two LSL webinars on our Brain Products Academy webinar channel*

Because our products are so actively involved with LSL, we have gathered some experience over the years. We have not only done our own test runs, but we have also listened to feedback from our users. While LSL indeed offers a lot of possibilities and typically makes data synchronization across multiple devices and modalities a lot easier, there are still some common mistakes that can be made, which can taint the experience. In this article, we provide potential solutions to such common mistakes.

Different latencies between hardware and software streams

Depending on your expectations, the first point may not be an issue at all, but rather just a good-to-know fact. When you connect to different data streams, each sample of each stream has a specific time stamp. Based on these time stamps, the samples can be synchronized between streams. However, what might often be neglected is that it can take time until the sample is ready to be given the time stamp and to be published to the network. This is the case for hardware-based streams, where such delays depend on internal buffering, wireless transmission latency, the operating system, and the device driver.

These latencies can also be affected by the settings, such as number of channels and sampling rate. The latency may only be a few milliseconds, but can also go up close to 100 ms, depending on the setup.

When using hardware triggers that are sampled directly with the EEG in the same amplifier, this latency is not an issue because it is the same for event markers and data samples. However, if you want to synchronize different hardware streams, or even use software-based markers to indicate events, the different latencies may have a big effect on the results, especially since software-based markers are published immediately without any hardware-related delays.

We have discussed this topic before in this blog article and this press release article, where we also provide some ideas on how to measure the latency and show some examples for different amplifiers and settings. This recent publication from the LSL authors also mentions this issue and provides other ideas on how to find out the latency for your specific setup.

Another option to deal with this latency is to deal with the correction directly in the LSL connector. This only works if the latency is known, is very stable, and there are not too many different options that can affect it. The first LSL connectors to really correct for the latency are the X.on App for Android™ and the X.on App for PC. Here, the stable latency caused by the Bluetooth® transmission is known and the time stamps are automatically corrected for this value. Now, this does not make the transmission magically faster: for a closed-loop system, the same delay due to the Bluetooth® transmission will still be there, but for synchronization purposes, the remaining latency (mainly caused by the wireless transmission latency) is only in the range of a few milliseconds.

The main message is: if it is important for your experiment to know about this latency, it should be measured before starting to record data. Once measured, the setup should not be changed afterwards. Since the latency is usually a stable offset, it is easy to correct for it in further processing steps. Theoretically, this can also be done directly when recording, but in this case, it may be difficult to undo.

Manual operating errors

Especially when connecting to multiple streams simultaneously, it is important to carefully adhere to the correct steps of events necessary for connecting to and recording all streams without losing any information.

For example, when you want to start a multi-stream recording in LabRecorder, you need to make sure that all those streams are already present in the network and streaming data. Sometimes, LabRecorder takes a while to connect to the streams, so it is always a good strategy to give it enough time. Otherwise, markers that are sent from your stimulus presentation software could be missed (or get a negative time stamp) because LabRecorder was still busy connecting to all the streams and starting to write data to the XDF file.

Whether a stream is constantly available in the network also depends on its source. The TriggerBox Plus, for example, works this way: once the network settings are configured correctly, the stream should be visible in the network. Presentation®, as another example, makes it easy to provide a valid stream to the network as soon as the program is started, whereas other tools usually open and close the stream at the beginning of a script. Worse, it can even happen that multiple instances of an LSL stream appear in the network if the script is prematurely terminated, and thereby the LSL streams are not closed properly. A good strategy is to always correctly open and close streams, and account for enough time so streams can be created, streamed, and connected to.

On the other hand, closing the stream before ending the recording can result in broken XDF files that might render your recordings useless.

Streams cannot be found in the network

One of the most common issues people encounter when using LSL is that, for various reasons, streams cannot be found in the network. For example, when using LSL Viewer or LabRecorder the desired stream name simply does not show up in the list of available streams. Some potentials troubleshooting steps include:

- **First, make sure that sender and receiver really are in the same network.** Especially when using WiFi, it can happen that devices connect to a different network automatically without you realizing it.
- **Make sure that IP addresses and ranges are correctly configured.** For example, when using a wired connection via a network switch, it may be necessary to assign IP addresses manually. This is the case if there is no DHCP server available in the network that takes care of assigning IP addresses. If IP addresses must be assigned manually; make sure to give each device a unique address in the same subnet. For example, one can have 192.168.1.42 and another one 192.168.1.43.

- **A misconfigured firewall often causes problems as well.** Depending on your setup, it may be necessary to allow access for individual programs for both sending and receiving LSL streams. Please be careful when looking for the respective program and make sure to select the correct one. There may be many different programs with very similar names, so it can happen easily that the names get mixed up (for example, when using Python tools with multiple Python installations on one computer).
- **If you are still experiencing problems when looking for streams, you can try to edit the *lsl_api.cfg* file to increase the chances of finding LSL streams.** Sometimes it can take a while for a stream to be found in the network and it can happen that streams drift in and out of focus, changing from visible to invisible randomly. This can become frustrating when you want to connect to multiple streams simultaneously. A simple solution is to make use of the aforementioned configuration file. For more information on *lsl_api.cfg*, please read the documentation here. You may not have the config file on your computer. If you do not, you can download the file from the above link (or copy and paste the text and save the file accordingly). There are three locations where you can save the file in which it will be found by LSL-based programs: (1) globally, (2) in a user-based folder, or (3) directly in the programs folder. You can check where these folders need to be in the description above; it depends on the operating system and your user settings. User-based settings override global settings, and program-based settings override user-based settings.

  The easiest way to make streams appear in your receiving programs seems to be simply adding all the IP addresses in the list for “KnownPeers”. Just write each IP address of every LSL stream’s source in the list enclosed by the “{}” brackets and separate them with commas. Of course, there are more things you can adjust in the *lsl_api.cfg* file. You can find more information in the documentation. As an example, this is how such an entry might look in the config file:

  ```
  [lab]
  KnownPeers = {192.168.1.42, 192.168.1.43}
  ```

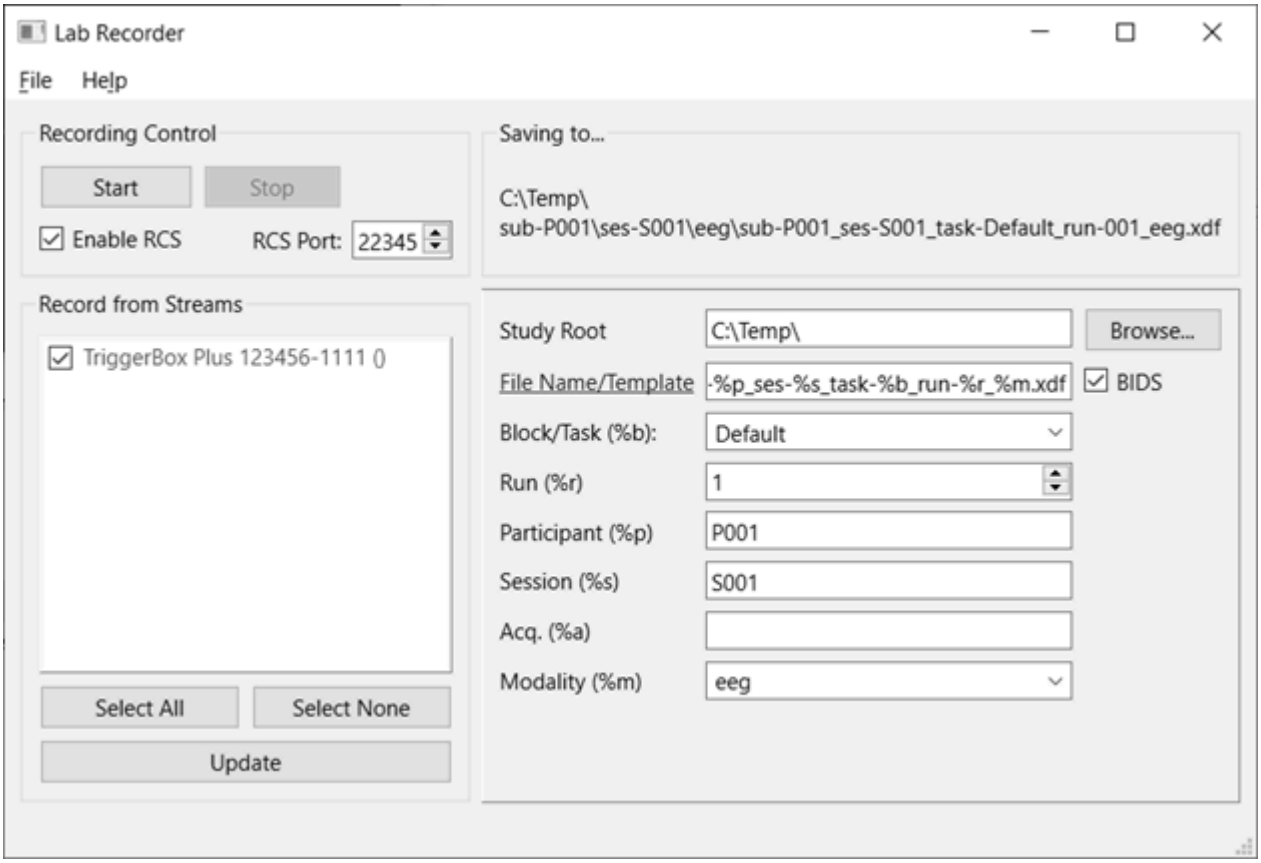
- **For LabRecorder, it can also help to provide the names of the required streams in *LabRecorder.cfg* (located in the program folder).** Here, you can list all the stream names that have to be present for the recording. If they are not present, or cannot be found, they show up in red colour:

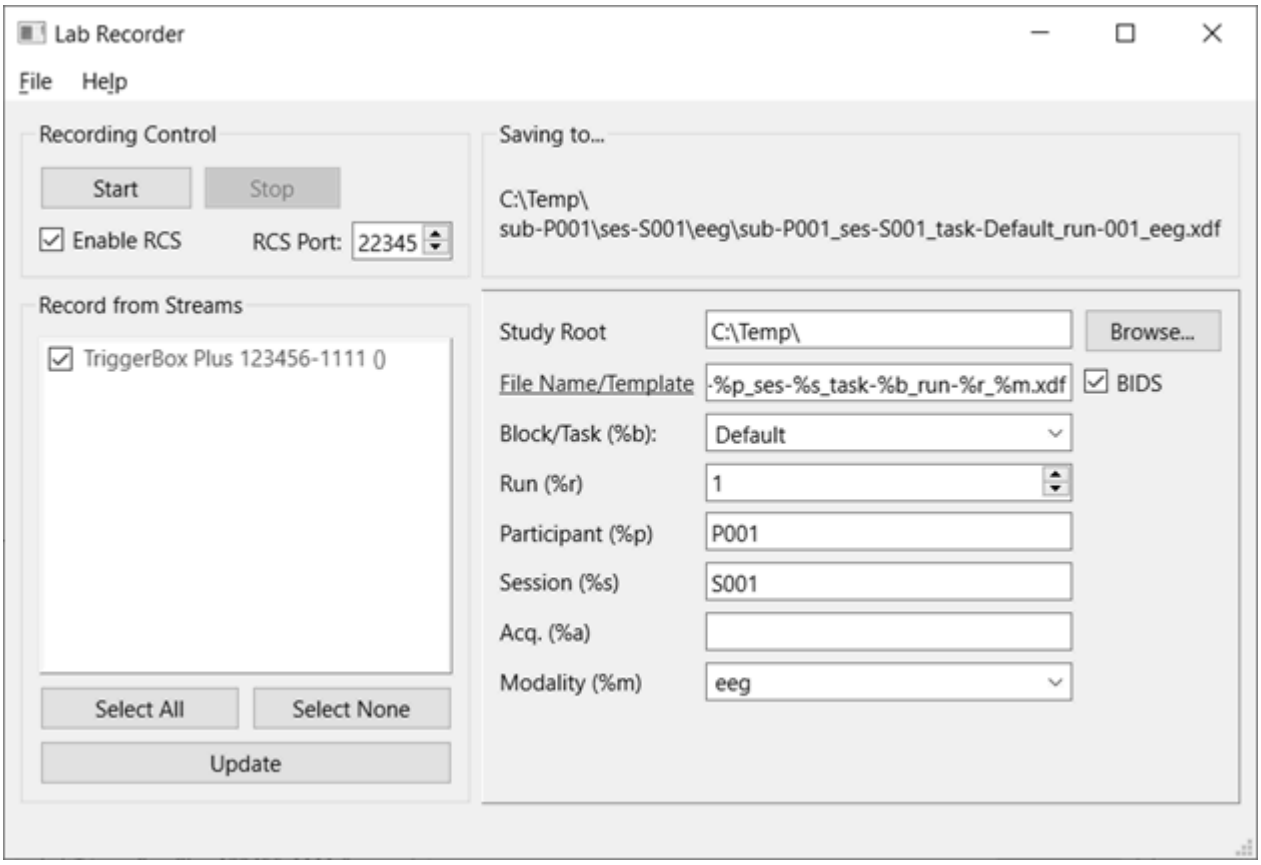
  ```
  ; === Required Streams ===
  ; This is optionally a list of streams that are required for the recording;
  ; a warning is issued if one of the streams is not present when the record button is
  pressed
  ; The syntax is as in: RequiredStreams = "BioSemi (MyHostname)","PhaseSpace
  (MyHostname)","Eyelink (AnotherHostname)"
  ; where the format is identical to what the LabRecorder displays in the "Record from
  streams" list.
  ; There must not be any spaces within the parentheses.
  ```

RequiredStreams="TriggerBox Plus 123456-1111 ()"

In this example, the name "`TriggerBox Plus 123456-1111 ()`" is comprised of the name of the device and its serial number. The bracket can remain empty if the `Hostname` is not included in the stream name. LabRecorder will search for this stream name at startup.

Once the stream is available (press "Update" to refresh the list), the stream will change the colour to green.





Importing XDF files

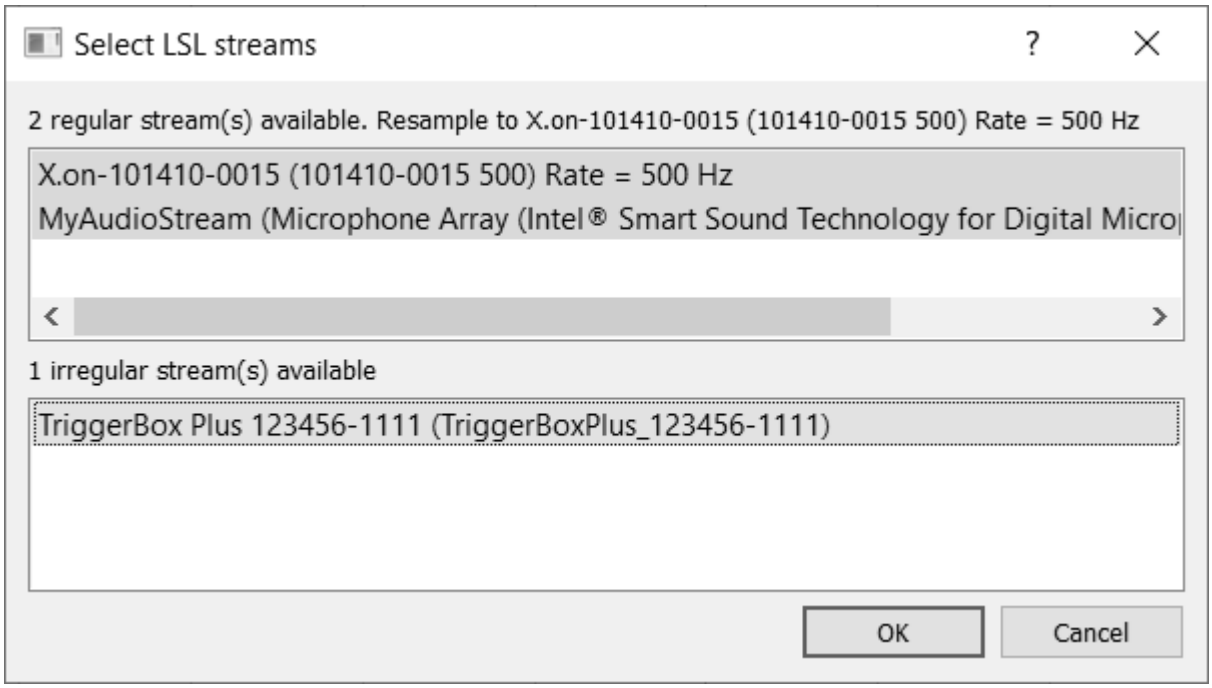
LSL has an associated file format, the eXtensible Data Format (XDF). If you want to record LSL streams, XDF is usually the obvious choice. There are a few things to keep in mind, especially when you are used to other file formats, in particular the BrainVision Core Data Format (BVCDF, the file format that is used by BrainVision Recorder, which consists of .eeg, .vmkr, and .vhdr files).

The main difference is that XDF keeps pairs of samples and time stamps, whereas BVCDF assumes a fixed and stable sampling rate and therefore has constant inter-sample intervals, which makes continuous time stamps unnecessary.

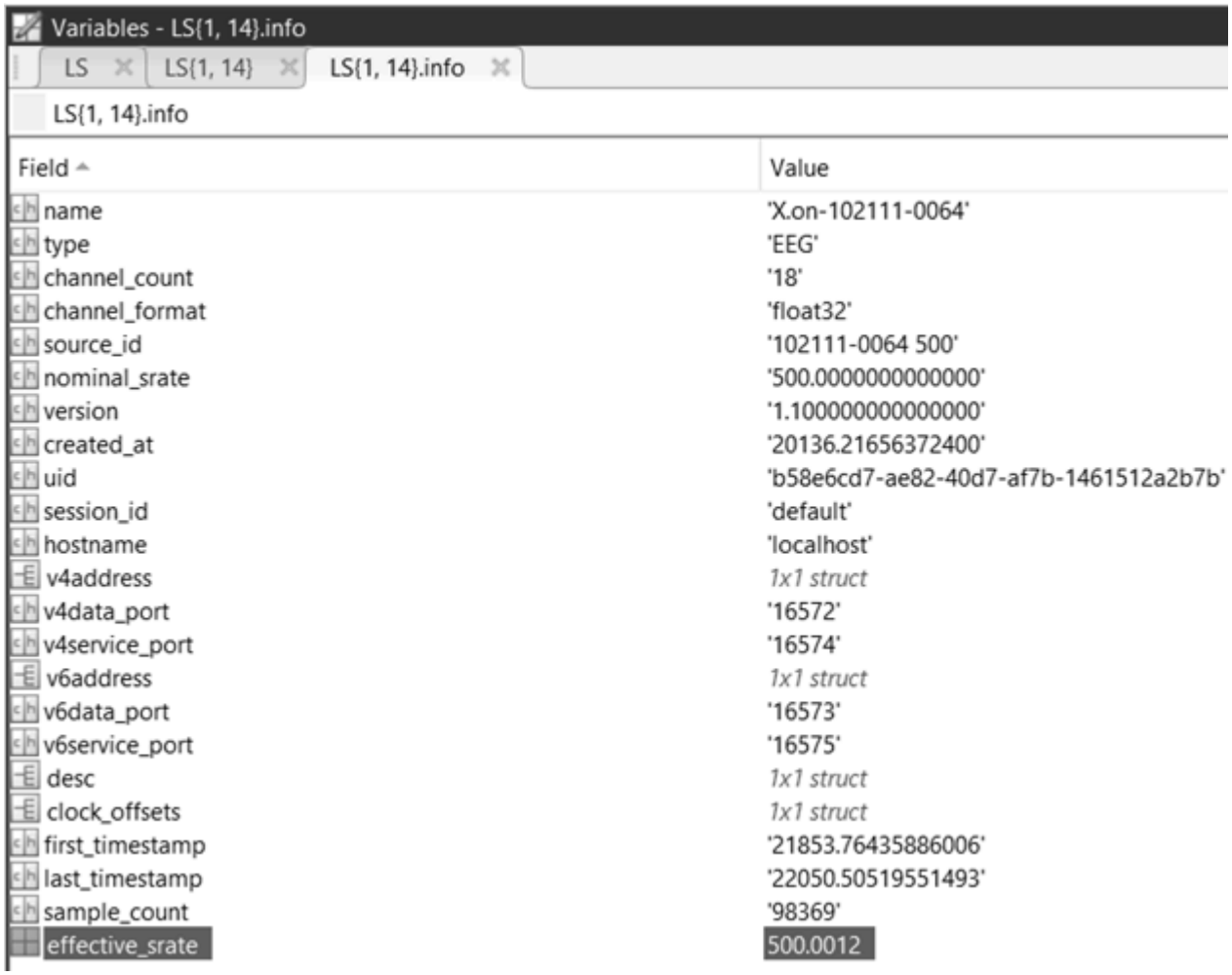
In XDFs, these intervals may not be so constant. In fact, it might also be that there are small or even larger gaps. That said, when importing such files to a file format without time stamps (BVCDF), these irregularities are no longer visible. Although it is the best strategy to avoid these irregularities altogether, the next best strategy is to detect them when they happen. There are some ways to do that:

- **Always monitor data online when streaming.** If you always observe your data, missing samples can usually be spotted. This can be done by special signals that count the number of samples which can be integrated in the LSL connector and might have to be selected (e.g., “Sample Counter in EEG” for the LiveAmp LSL connector). If this signal is not growing one-by-one, you know that something went wrong. Another way is to simply check if the data is really happening at the very moment: move the amplifier if it has accelerometer sensors integrated and see if the signal matches the movement, or clench the teeth when streaming EEG and see if the noise happens when and where it is expected. Another possibility is to apply display filters that cause oscillations at discontinuities, for example, when samples are missing, and the filter must settle anew.

  Our own LSL Viewer is a great way to monitor signals. The newest version (≥1.0.0) even allows displaying multiple continuous signals at the same time, by resampling to the first stream selected:



- **Check your saved data and verify that the effective sampling rate ("effective_srate") is as close to the nominal sampling rate as possible.** The "effective_srate" can be found in the *info* field of every LSL data stream. If you set the LSL connector of your amplifier to 500 Hz, this constitutes the nominal sampling rate. If the effective sampling rate is not 500 Hz, but instead 490 Hz, you know that you had a problem during recording. Some deviation around the ideal value is to be expected, but it should remain in the range of under 0.01%. To get an idea: a deviation of 1% from the nominal sampling rate would mean a drift of 36 seconds per hour. An effective sampling rate that is off significantly can mean that there are occasional missing samples or that there are one or more large gaps. Depending on the nature of the recorded data, it may be useable to some degree, but it should be handled with care.



- **You should also be aware that XDF import tools usually apply a dejittering algorithm.** For example, in the MATLAB® *load_xdf.m* function, this is activated by default. If you do not want to use it, you must set *'HandleJitterRemoval'* to *False*. It is also activated in Python's pyxdf load function. While the jitter removal is generally the best way to go, it can cause problems when the data recording was not as smooth as expected. For example, it can introduce a fake drift in your data, and the smoothed time stamps (which alter the actual time stamps as they were recorded) make it harder to spot problems in the recording. That said, it is always a good idea to look at the data at least once with jitter removal deactivated.
- **When importing a file with many streams, the sequence of streams can vary between recordings.** What was once stream #1 can be stream #8 in the next recording. Therefore, always make sure to open the correct stream. There are unique entries in the metadata of each stream, indicating the name, source, and type of each stream.

> ***Note:*** *If you have found a way to control the sequence of streams, please do let us know!*

More network troubles

The network component plays a big role in the world of LSL. It goes without saying that a bad network setting can lead to bad experiences when a lot of data needs to be transmitted.

Problems that might affect the performance negatively and potential steps to avoid them are:

- **Poor wireless connection or network load near capacity.** If possible, always try to use a wired connection, ideally, in a dedicated local network without connection to the general internet. Otherwise, you can never exclude large data transfers due to updates, or someone streaming data with a high bit rate. After all, the network has a limited capacity and ideally you do not want to share it with anyone else.

  It may also be possible to tune the settings in the *lsl_api.cfg* file to these values (see this publication):

  ```
  [tuning]
  TimeProbeMaxRTT = 0.10072
  TimeProbeInterval = 0.01073
  TimeProbeCount = 1074
  TimeUpdateInterval = 0.2575
  MulticastMinRTT = 1.076
  MulticastMaxRTT = 30
  ```

- **Additionally, restrictive firewall settings or access restrictions** within an existing network (e.g., in your company or university) may make the connection slow and/or increase the problems when trying to find and connect to your LSL streams. Again, try to create a dedicated network with a proper network switch, router, or even create a personal network with your smartphone if there are no other options.
- **Using WiFi in the 2.4-GHz bandwidth may cause interference with your Bluetooth® devices** because they operate in the same range. If possible, connect to a 5-GHz network. For some routers, it is possible to create separate Service Set IDentifiers (SSIDs) for each bandwidth, which makes it easier to select the correct network.

Differences between BrainVision Recorder and LSL Connectors

Another potential hurdle is the difference between operation of BrainVision Recorder and LSL Connectors. Our standard acquisition software offers great flexibility via a wide range of options to configure each of our amplifiers. However, there are some quirks that our customers might have gotten used to. Handling these idiosyncrasies differently might be confusing to some.

- As mentioned before, there is only one time stamp/marker at the beginning of the recording in the standard BrainVision file format. However, in XDF, there are always pairs of samples/time stamps.
- In BVCDF there is only one continuous stream. In XDF there can be many different streams, all with different sampling rates. Therefore, it is not a simple task to import XDF files to BVCDF. However, it is possible. Just keep in mind that resampling to a common sampling rate is necessary first. MNELAB, which we recommend as an easy way to import XDF files and convert them to BVCDF, provides this feature since version

0.8.0. For more information on how to convert XDF files to BVCDF, please read this support article.
- One characteristic of BrainVision Recorder is that the value ‘0’ is not shown as an event. In LSL, this is not the case. While this provides more information to the user, it can certainly cause confusion when triggers do not look the same between two different acquisition tools.

Failure resilience

LSL typically handles connection problems very well. If an LSL outlet is disconnected from the network (e.g., due to an empty battery or connectivity issues), the inlet will try to reconnect with the stream. Apart from a warning, the user might not even realize that there was a problem while recording. However, there will at least be a gap in the recorded data during the time the outlet was not actively streaming. When importing the data via a GUI, one might miss that there is a gap as it might be automatically concatenated.

Again, it is in the user’s own interest to check the recorded XDF file for such gaps and account for it when loading the file, when working with the data, or when exporting it to another format, such as BVCDF.

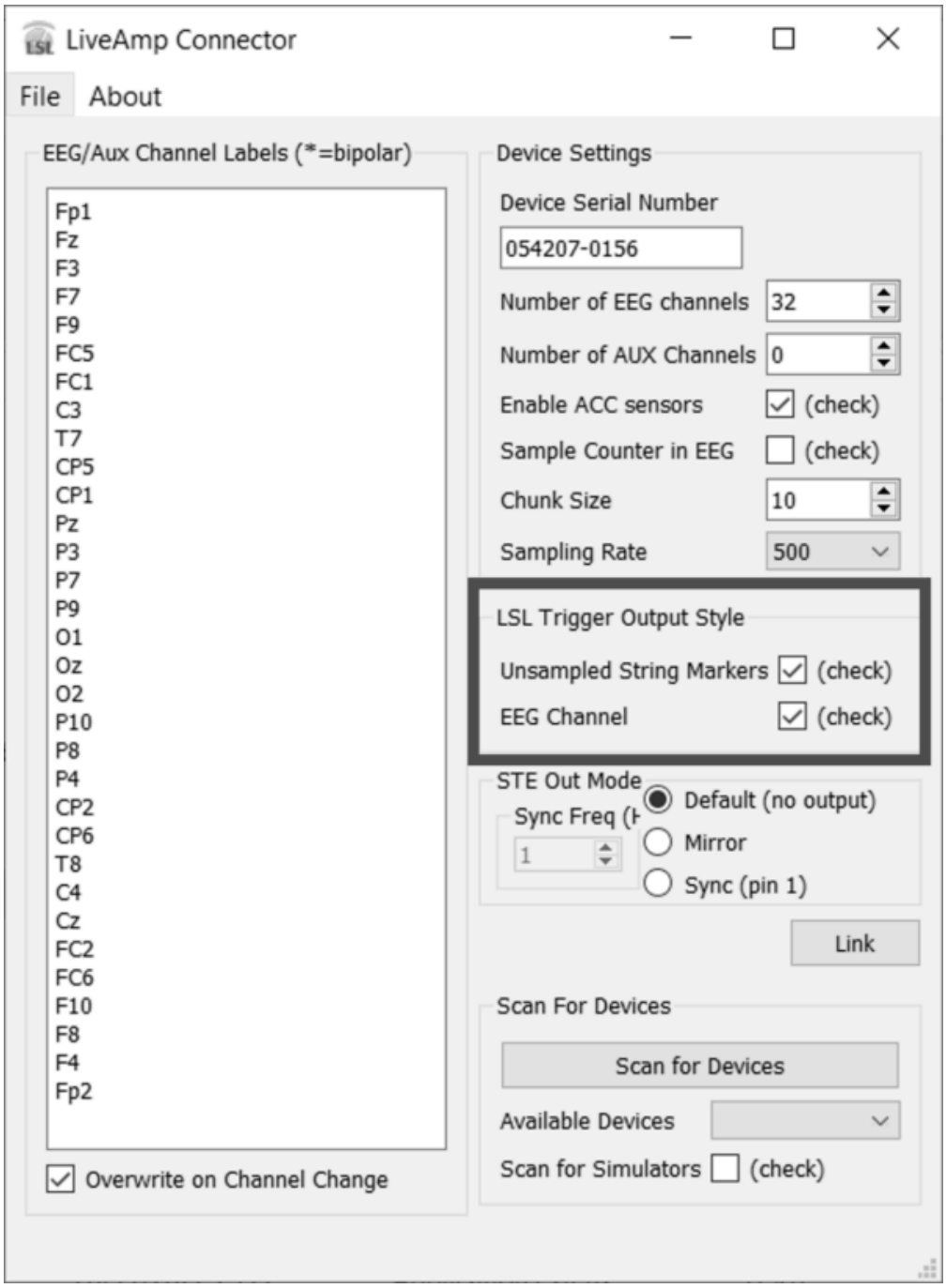
Unsampled triggers versus EEG channel triggers

There is one more common problem, or rather misunderstanding, that involves the different options for streaming hardware triggers in Brain Products LSL connectors, called “LSL Trigger Output Style”.

Hardware triggers can either be sent as “Unsampled String Markers” and/or in an additional “EEG Channel”. It is highly recommended to use the “EEG Channel” option if timing precision is important. When doing so, triggers are sent in an additional channel with samples for every time stamp (a constant ‘-1’ except for when a trigger is happening, then the actual value of the trigger is sent). This means that triggers can be properly dejittered similarly to other regular signals.

If only the “Unsampled String Marker” option is selected, such dejittering is not possible, which results in trigger values that have an approximately 10x higher jitter. The advantage of this option is that it is easier to work with these markers, as they show up as numbers in online streams and can be parsed without having to check a whole channel. They will only have as much recorded sample/time stamp pairs as there were triggers.

Of course, it is possible to select both options and use the positive features of both at the same time.



There are ways to decrease the jitter directly in the LSL connector and we are working on including these improvements in future versions. Stay tuned for updates regarding future updates.

We hope this collection of tips & tricks is helpful to you and that this article makes the use of LSL in your experiments even easier. Do not hesitate to contact us with any kind of feedback, either to point out missing issues, better solutions, or to just let us know about your experiences.