MiRA::

FLUX:: Immersive

2025-06-02

Table of contents

MiRA::

25.01 version

A User Guide

Written by Lorcan Mc Donagh & Jean-Loup Pecquais

with contributions from: Gaël Martinet, Vincent Carlier, Florie-Anne Lafaye, Hugo Larin & Nicolas Erard.

Legal Information

First edition of FLUX:: MiRA User Guide published in January 2025


```
FLUX:: MiRA is published by
Harman Professional, Inc.
8500 Balboa Blvd
Northridge, CA 91329
United States
```


Product Page | Shop Page

Part I

Welcome to MiRA::

Product Page | Shop Page

First of all, thank you for acquiring *FLUX::MiRA::*. We hope it will provide you with new levels of insight, understanding, and control over your audio signals. Our goal was to deliver the most comprehensive real-time audio analysis tool, and to make the whole process of audio analysis powerful and intuitive for beginners and professionals alike. Our real-time audio environment provides easy access to some of the most advanced audio analysis algorithms currently available.

FLUX::MiRA:: is a real time audio analysis software that allows sound engineers in various fields of expertise to visualize and understand the frequency and time domain characteristics of audio signals in real time.

***FLUX::MiRA::* maintains the highest audio quality throughout the entire signal flow.**

Part II

FLUX::MiRA::'s Heritage

Behind *FLUX::MiRA::* is the expertise of FLUX:: SE, a leading developer of audio software and technology. FLUX:: SE has a long history of developing high-quality audio software, and *FLUX::MiRA::* builds on this legacy of excellence. The software is designed to meet the demands of professional audio production, and to provide users with the tools they need to analyze and understand their audio signals.

Part III

About This Guide

This guide has been written for practitioners already working in audio production yet new to the *FLUX::MiRA::* software environment. It is also intended to be read as a practical introduction to audio analysis for those who are new to the medium and coming to it through *FLUX::MiRA::*. Of course, there is plenty more knowledge to be had in the field of audio analysis and the technology behind it, which will strengthen your understanding and decision-making.

We strongly suggest spending the time to read through this guide before starting on your first major analysis and keeping it on hand during the process.

Getting started

In the following section, we will explain how to install the application using the FLUX::Center and how to redeem and activate your licenses.

We will also explain all the differences between the **Session**, **Studio**, **Live** and **Ultimate** licenses.

Furthermore, we also feature a quick start section to help you to be all set up in no time.

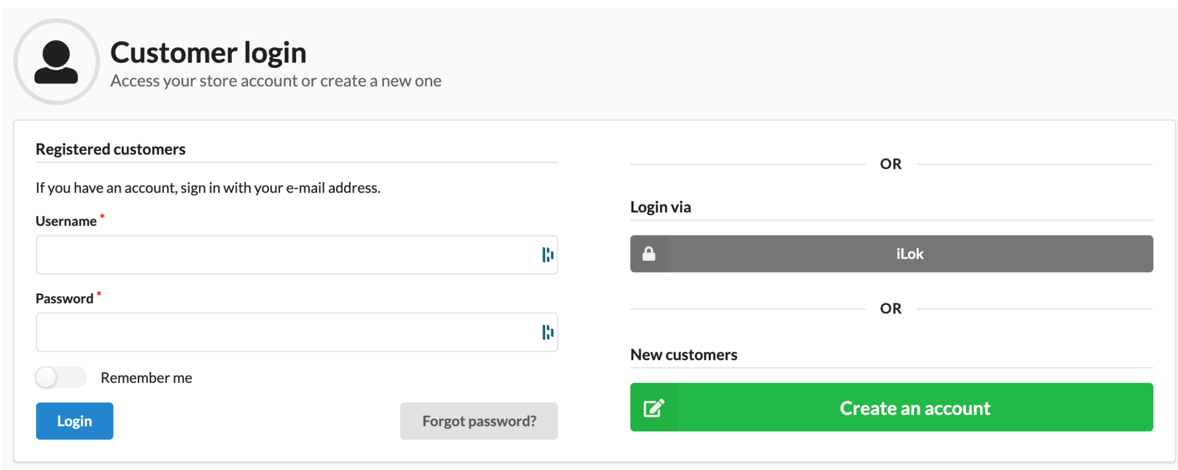
1 Installation and activation

1.1 How to Install FLUX MiRA::?

FOUR (4) STEPS:

1. Create an account on FLUX website
2. License code redeem
3. Software license activation
4. Download and installation

1.2 Create an account



Create an account on the FLUX website by clicking on the previous link.

1.3 iLok User Account

To activate licenses:

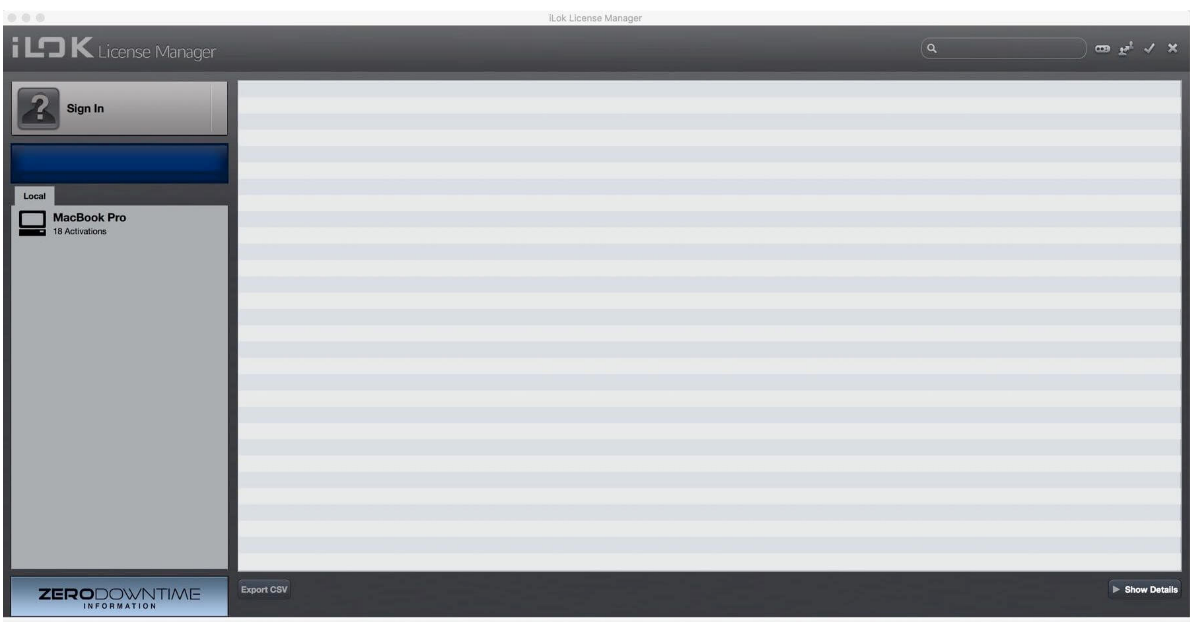
- An iLok user account is required.
- An iLok USB key is optional.

FLUX:: uses the iLok license management system to deliver software licenses to users. If you don't have an iLok account yet, please create a free iLok account at http://www.ilok.com and download the iLok license manager. *FLUX:: MiRA::* includes two (2) activations linked to your user account. Having two activations gives you the possibility of a fixed license on one particular machine and a portable license on an iLok USB key if you own one.

> **Note**
>
> Cloud license is currently not supported.

1.4 iLok License Manager

If you have redeemed your software license or completed your purchase process, your license will automatically be delivered to your iLok account.



For new iLok users, the first step is to download and install the iLok license manager available on the home page of the iLok website. When your user account is successfully activated and the iLok license manager is correctly installed, you can start the license manager software and log in to your iLok user account.

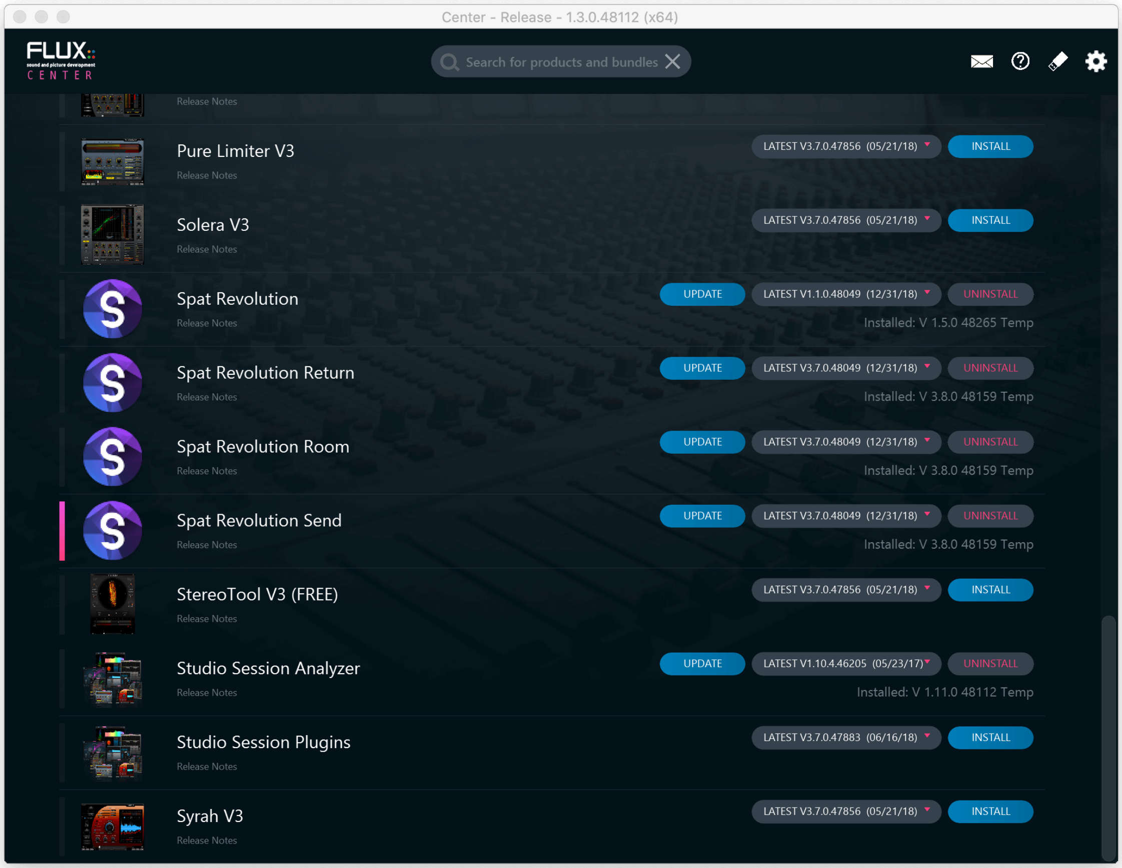
1.5 Transferring license

Pressing on the sign-in button will allow you to connect to your account. After Logging in, you are now ready to transfer any licenses to a computer or to any iLok USB key if you happen to have one. The process of transferring a license is as simple as dragging the license from the Available tab to your Local Computer *(or iLok key)* on the left side.

Simply drag your license to your Local Computer or on an iLok USB key. You are now set!

> **Note**
>
> If you require further information about iLok and managing licenses, please refer to iLok.com website.

1.6 FLUX:: Center

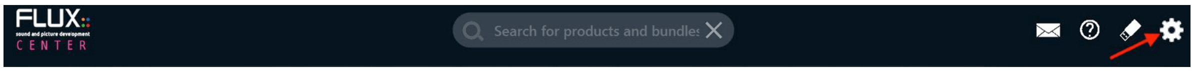


The next step is to get the installers for the FLUX:: products you are licensed for. All the software and plug-ins from FLUX:: are available via our FLUX:: Center software. This is a Mac or Windows application we have created to help keep your FLUX:: products up to date and to give you a clear overview of what you have installed. Firstly, please visit the download section of the FLUX:: Website to get the installer for the FLUX:: Center application.

On this page, you will find a macOS and a Windows 64 bits, as well as legacy versions for older operating systems. After downloading and installing, you can open the FLUX:: Center applications to begin the process of installing the *FLUX:: MiRA::* software.

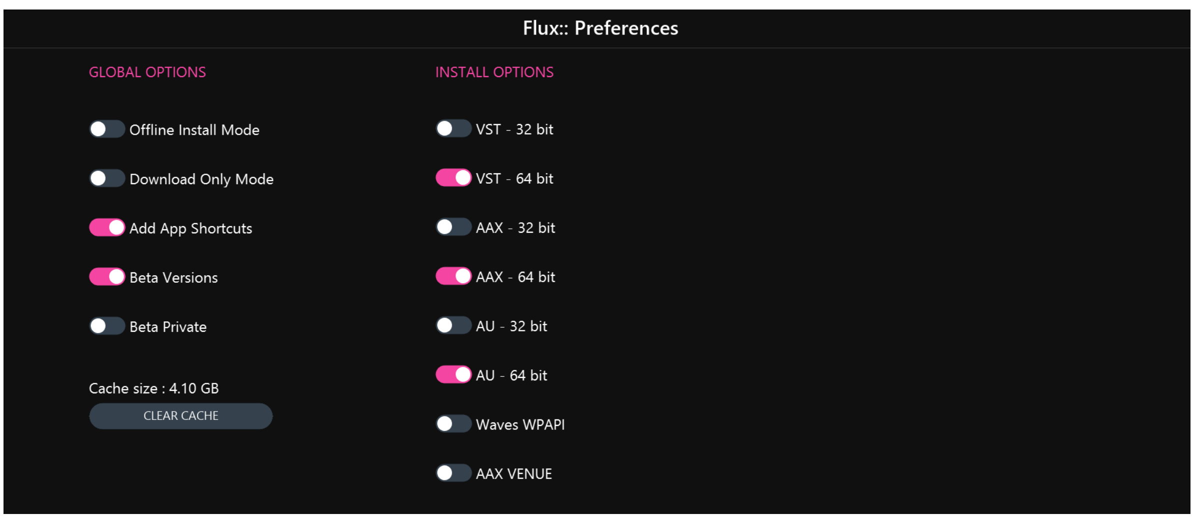
> ⚠ Warning
>
> Authentication is required at the launch of FLUX:: Center. This is the login details of your FLUX shop account, which allows you to see only your products licensed for (temporary or permanent).

1.7 Center Preferences



When you open FLUX:: Center you will see a page listing all FLUX:: products available for you to install. You will also find information about which version you have currently installed on your system and which new versions might be available for you to update. You can select versions to install - or uninstall if necessary - using the pull-down menus. If you would like to access more installer options such as your preferred plug-ins format, please click on the gear icon at the top right of the header area.

1.8 Center Preferences and Options



This preference page will allow you to choose various installation options, such as preferred plug-in formats for your system. Choosing your format and returning to the main page by pressing the OK button will show all your install options for software and plug-ins based on the desired formats chosen.

If you would like to be closer to the most current development cycles of the software, you can enable the Beta Version option. This will give you access to a special set of software installers from the pull-down menus on the main FLUX:: Center page. Beta versions are the new builds that are still under development but may contain useful bug fixes and new features. If you find that a beta version is not stable enough for you, then you can always roll back to a stable release version at any time through the FLUX:: Center installers. Note that these versions start with a `B` whereas official releases start with a `V`.

Licenses and MiRA:: Versions

MiRA:: is available in three different software versions: MiRA:: Session, MiRA:: Studio and MiRA:: Live. MiRA::Ultimate is a bundle of Live and Studio versions.

The main differences are:

	MiRA::Session	**MiRA::Studio**	**MiRA::Live**
Reference Inputs/Outputs	Up to 2 channels	Up to 24 channels	Up to 24 channels
Multi-channel speaker arrangement support	Stereo	3D immersive support	up to 24ch frontal systems
System tuning Inputs/Outputs	None	None	Up to 24 channels
Sample rate	Up to 96 kHz	Up to 384 kHz	Up to 384 kHz
Scopes	Magnitude Spectrum, Spectrogram, Nebula, Vector Scope, Wave Scope, True-Peak Metering, RMS Metering, Loudness Metering	Session + Nebula 3D and Metering history + Off-line metering	Session + Transfer function, Impulse response and Leq

2 Quick start & typical setup

MiRA:: can be used in two ways: to analyze the content of an audio stream, or to generate captures from measurements to obtain the transfer function of an audio system or equipment.

> **!** Important
>
> Captures are only available with a Live license.

The following instructions will guide you through the MiRA:: setup process.

2.1 Acquiring audio

There are two main solutions to input audio streams into MiRA::. First, we can use MiRA:: with an audio interface. The audio interface then receives the incoming audio signals to analyze.

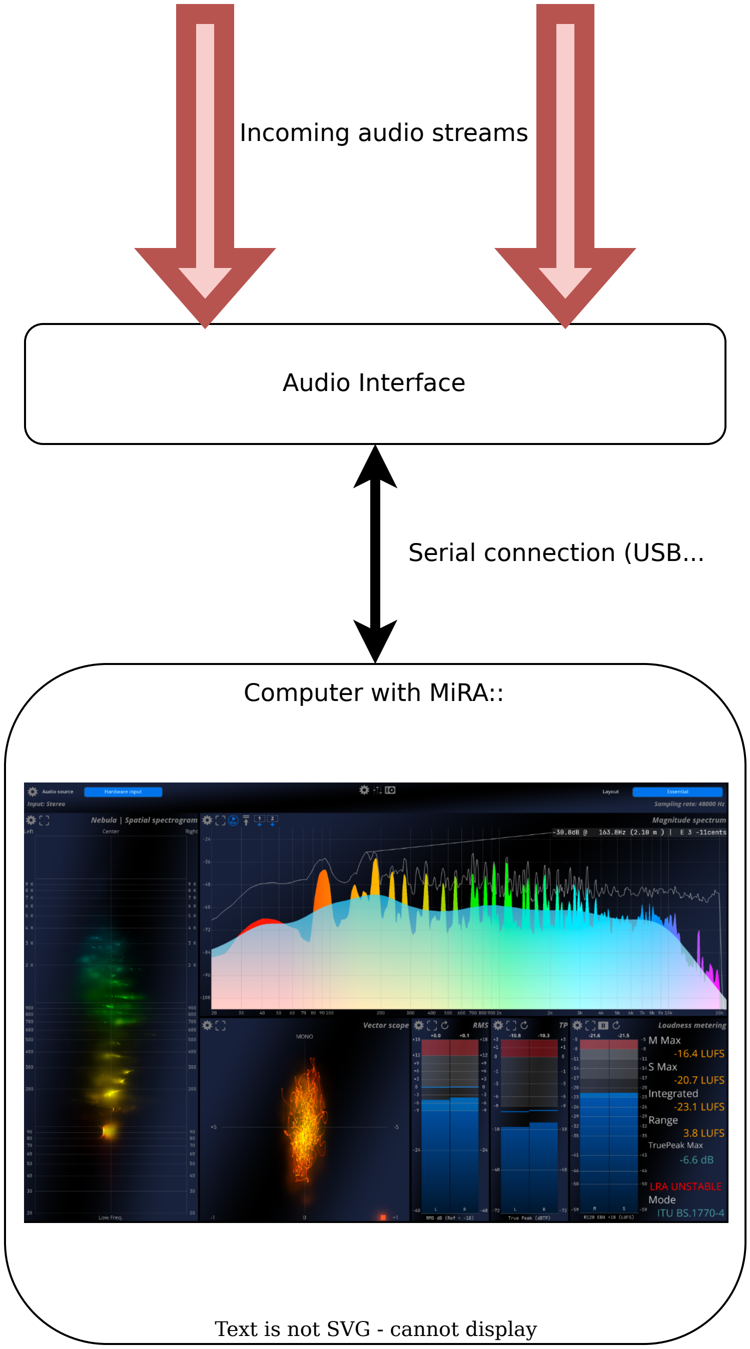


Figure 2.1: MiRA:: with an external audio interface

In many cases, having to rely on a dedicated audio interface is inconvenient. To streamline the workflow, we have developed a technology called **SamplePush**. The principle is very simple: SamplePush is a protocol that allows you to stream audio over a network using auto-discovery. Two of our products implement this protocol: the **SampleGrabber**, which comes with MiRA::, and **SPAT Revolution**.

SampleGrabber is a plug-in, available in most common formats, that allows streaming audio from a digital audio workstation (DAW), or a plugin host, to MiRA::.

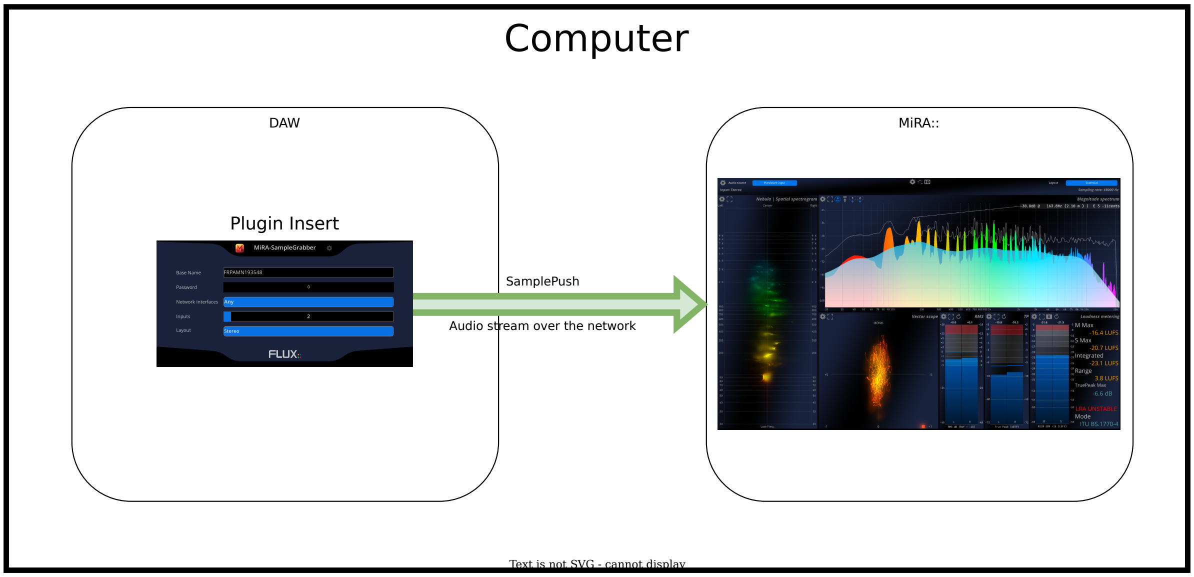


Figure 2.2: MiRA:: with SampleGrabber

2.2 Selecting an audio source

Most of the factory layouts include the info header bar that makes it easy to choose the input source.

Figure 2.3: Header bar with audio source selection on the left

Some layouts do not have a header bar. The audio source selection is still accessible from the top menu, `MiRA>IO Settings`.

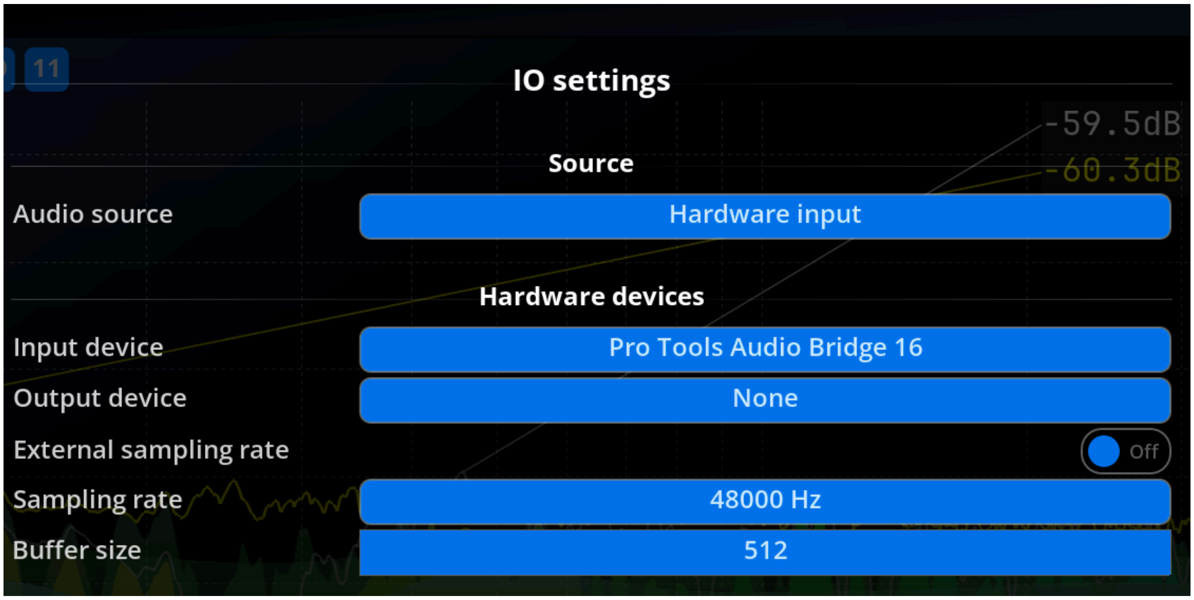


Figure 2.4: Audio source selection from the Input/Output settings menu.

This menu also gives access to the sample rate settings.

2.3 Typical setup

2.3.1 Home / Project / Mobile studio

In this type of environment, we typically use a single computer for all tasks, so both a DAW and MiRA will be running concurrently.

In this case, using SampleGrabber is certainly the simplest option. Simply add it to the end of your master track to send your mix output to MiRA::

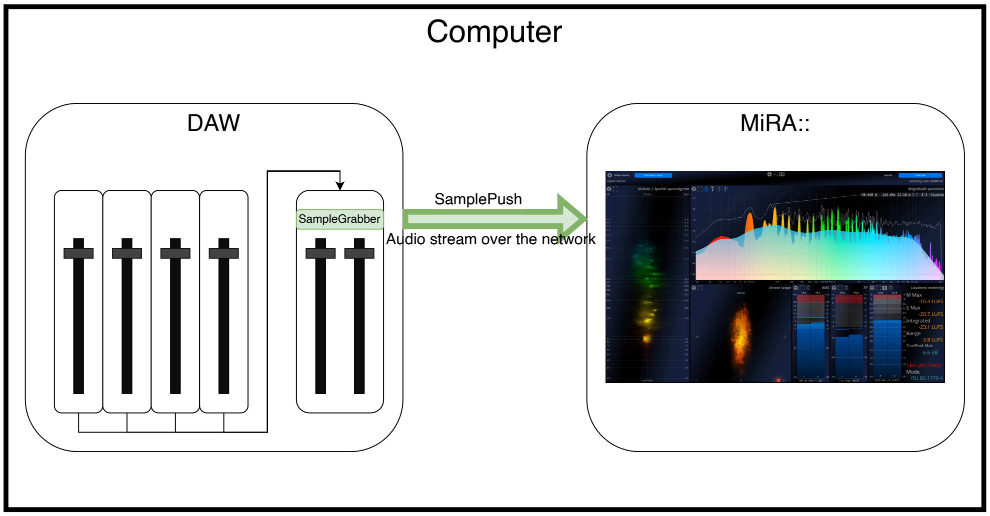


Figure 2.5: MiRA:: in a home studio context

2.3.2 Mixing and mastering studio

In larger structures, sometimes separating concerns by using several computers for different applications can be more prudent.

MiRA:: is not resource-intensive, so a mid-range desktop computer can handle its execution. Please refer to the system requirements page of this manual.

Even if MiRA:: runs on a dedicated computer, the SampleGrabber can still send the audio stream if both computers are connected to the same local network.

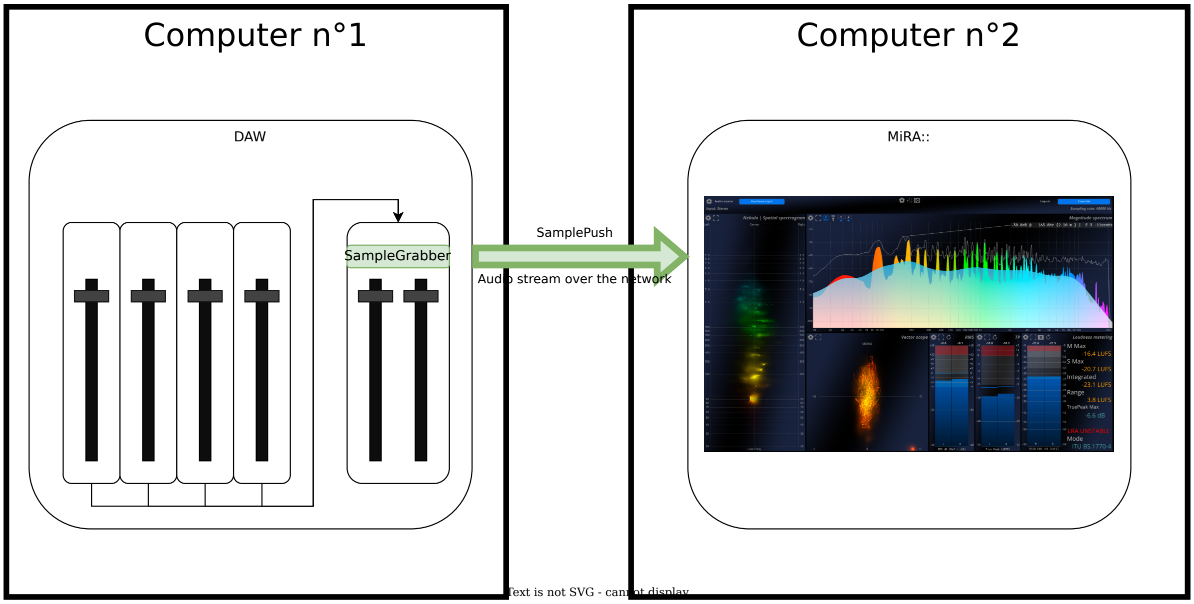


Figure 2.6: MiRA:: in a mix/mastering studio context

The SampleGrabber is capable of streaming up to 24 channels of audio if you want to monitor an immersive mix of some sort.

> **!** Important
>
> Immersive analysis is only available in MiRA::Studio version.

2.3.3 Autonomous mobile configuration

For system tuning and equipment measurement, having an autonomous and portable solution is very important.

An entry-level laptop can handle the task. The audio interface should have at least two inputs and outputs. For more information, please refer to the system requirements page in this manual.

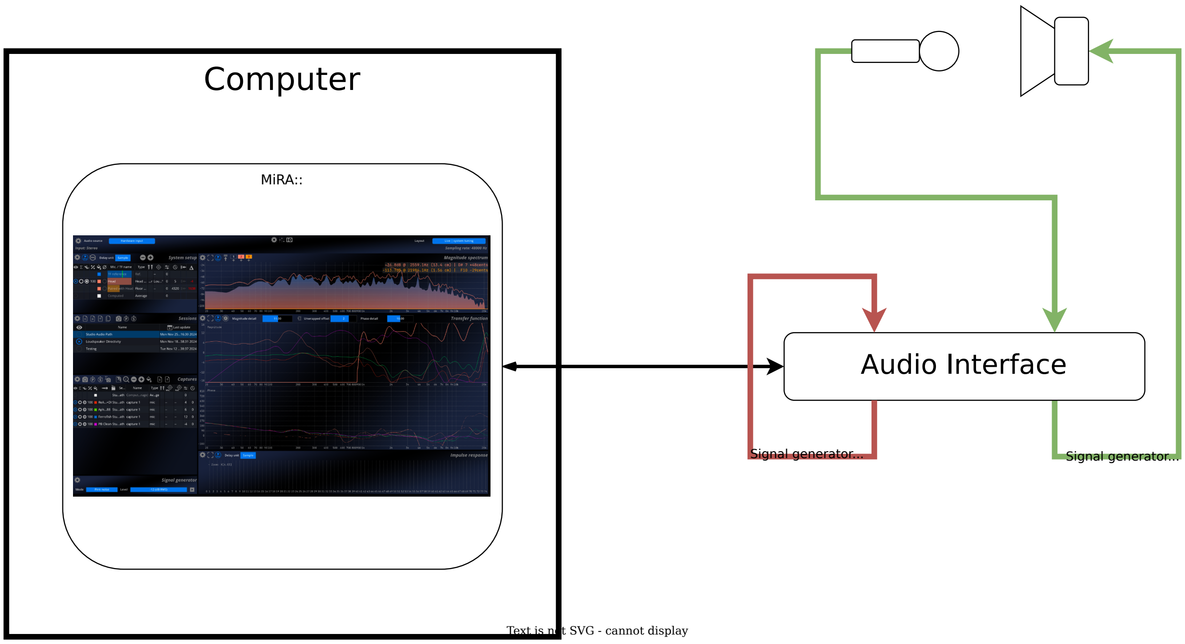


Figure 2.7: MiRA:: Setup for system measurement

2.4 Navigation

MiRA:: is an app with a modular user interface. Each “page” is called a layout. Each layout has several audio analysis tools that we call scopes. Each factory layout has a specific use case in mind, so it is important to know how you can simply and quickly navigate through them.

2.4.1 Using the info header bar

The info header bar is a UI scope that gives you quick access to several options, as well as a drop-down menu for selecting a layout from those available in the current workspace.

> **!** Important
>
> Some factory layouts don’t feature the info header bar. Refer to another navigation method in this case (see below).

2.4.2 Using the top menu

In the MiRA:: top menu, you can go to `View>Layout` to toggle a specific layout, or move forward or backward.

2.4.3 Using shortcuts

MiRA provides two primary shortcuts for navigating layouts:

- `TAB` loads the next layout
- `Shift+TAB` loads the previous layout.

3 Files and folders

3.1 Files types

MiRA:: uses two different types of files:

- .json
- .fcap

The *.json* files are the main files of *MiRA::*: workspace are saved in this file type.

To save a workspace, click on the "Save workspace" action on the "File" menu, or use the shortcut `Ctrl + S` on Windows, or `Cmd + S` on Mac.

.fcap represents an export of a session, available in the system analysis part of *MiRA*.

3.2 Preferences

The preferences of *MiRA::* are located on the following folder:

- `~/Library/Application Support/FLUX/MiRA` on macOS
- `C:\User\/...\AppData\Local\FLUX\MiRA` on Windows.

In it will be located three folders:

The "Captures" folder contains all your captures' files, made from the system analysis part of *MiRA::*.

The "Preferences" folder, contains 3 files: - current_state.json contains the latest workspace you are working on - users.json contains your saved software preferences - UI.xml saves your user interface preferences

A subfolder named *Shell* contains:

- history.txt - a history of the terminal commands

Part IV

Theoretical knowledge

4 Introduction to Audio Analysis

As a sound engineer, your primary role is to capture, manipulate, and reproduce sound to achieve the desired audio quality. To excel in this field, a solid understanding of audio analysis is indispensable. Audio analysis encompasses various techniques and tools used to examine, measure, and interpret audio signals. Our aim here is to provide an introductory overview of audio analysis, focusing on key concepts, tools, and techniques crucial for sound engineers.

Audio analysis serves multiple purposes in sound engineering. It **helps in identifying and resolving** technical issues, optimizing acoustic environments, enhancing sound quality, and ensuring consistency across different platforms and mediums. By analyzing audio signals, engineers can make informed decisions about equalization, compression, noise reduction, and other processing tasks.

4.1 Fundamental Concepts in Audio Analysis

Any signal can be either seen from its evolution in time or from its frequency content. An analyzer tries to extract relevant data from an **audio stream** to turn it into a meaningful **visual representation**. This leads to the two major families of analyses one can perform on an audio signal:

4.1.1 Frequency Spectrum

The **frequency spectrum** represents the distribution of energy across different frequencies in an audio signal. Tools like Real-Time Analyzers (RTAs) and spectrograms are commonly used for this purpose.

In a conventional digital system, audio material is captured, stored, transmitted, and reproduced as a sequence of values, which correspond to the amplitude variations of an electric signal at discrete points in time. Our ability to extract meaningful information from this raw data through either hearing or visualization of the signal curve is somewhat limited to emotional interpretation, which is extremely subjective.

Extensive studies have shown that first converting this data to a so-called frequency representation is extremely useful for a broad range of audio applications, as it is quite similar in principle to the human auditory system. A proper detailed explanation of the reasons behind this is well outside of the scope of this manual, so we will only hint at a few important characteristics of human hearing, namely its:

- Ability to recognize and isolate sounds based on their relative intensity or loudness
- Ability to identify a pitch and timbre (color, texture) for sounds that fall in this category
- Ability to distinguish sounds based on their actual or perceived location

A fundamental tool for transforming a time-based digital audio signal into a frequency-based representation, a.k.a frequency spectrum, is the discrete Fourier transform (DFT) and its derivatives, such as the Short-Term Fourier Transform (STFT) and Fast Fourier Transform (FFT). Basically, the DFT maps a signal to a set of amplitudes taken at equally spaced frequency intervals. In essence, one can see the DFT as a bank of many band-pass filters, with as many meters at the output of these filters.

4.1.2 Level Analysis

Level analysis is a fundamental aspect of audio signal evaluation, focusing on the measurement and monitoring of signal amplitude over time. This process involves tracking various level metrics, such as peak, RMS (Root Mean Square), and loudness units, to ensure optimal dynamic range and prevent distortion. Sound engineers use level meters to visualize these metrics, enabling them to make informed decisions about gain staging, compression, and limiting. Oscilloscopes and waveform analysis can also give some significant insight into distortions that may have happened on the signal.

4.2 MiRA: Advanced Audio Analysis Suite

MiRA equips sound engineers with a comprehensive range of **real-time audio analysis tools** designed to streamline workflows and enhance output quality. At the core of these tools lies FLUX's proprietary **Variable Q Transform** algorithm, which outperforms the classic FFT by offering both reduced computational load and superior data readability.

MiRA features **industry-leading** spectrum analyzers, spectrograms, true peak/RMS/loudness meters, oscilloscopes, and vectorscopes. The application also allows users to customize their workspace by arranging these tools to suit their specific needs. Each tool offers extensive settings for further personalization.

A unique feature of MiRA is its **spatial spectrogram**, a powerful tool designed to analyze the spatial characteristics of audio signals. This sophisticated tool generates a detailed map of the soundscape, enabling engineers to understand and manipulate the spatial distribution of audio elements with ease. This capability is invaluable for crafting dynamic and engaging audio environments, ultimately providing listeners with a more immersive experience.

4.3 Understanding Audio Signal Chains and the Role of Measurement Tools

At first glance, an audio signal chain is very much like a series of black boxes. As an audio engineer, you can trust your ears and the manufacturer's data sheets to assess the effects this chain has on the incoming audio. In a variety of cases, however, this is either simply impractical, not possible, or not precise enough. Such situations include live sound setups, recording setups, etc., where unknown factors, such as the venue's or studio's acoustic response, are a crucial part of the chain.

It is therefore necessary to resort to scientific measurement procedures and tools to obtain precise, trustworthy, and reproducible results. The main tools at your disposal for this purpose are transferring curve and impulse response measurements, which are especially designed for this task.

As with any measurement instrument, it is important to have a good grasp of its mode of operation as well as any possible limitations in order to use it most efficiently. Some knowledge of acoustic principles and notions of signal processing are naturally required as well. While this manual tries to cover the most typical use cases and points out common do's and don'ts, it obviously cannot replace either a good textbook or practical experience.

5 Exploring the Fourier Transform

This chapter provides an easy-to-understand introduction to the Fourier Transform and its derivatives. It explains the technology behind a spectrum analyzer, which will help sound engineers understand the capabilities and limitations of their tools.

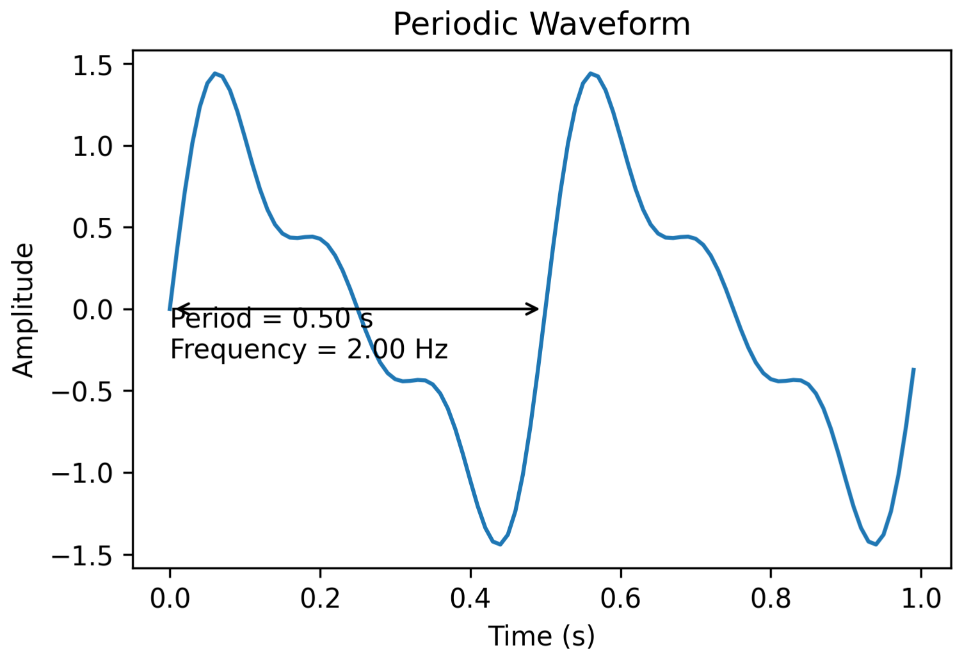
5.1 Introduction to the Fourier Transform

5.1.1 Fourier and the Dual Representation of Signals

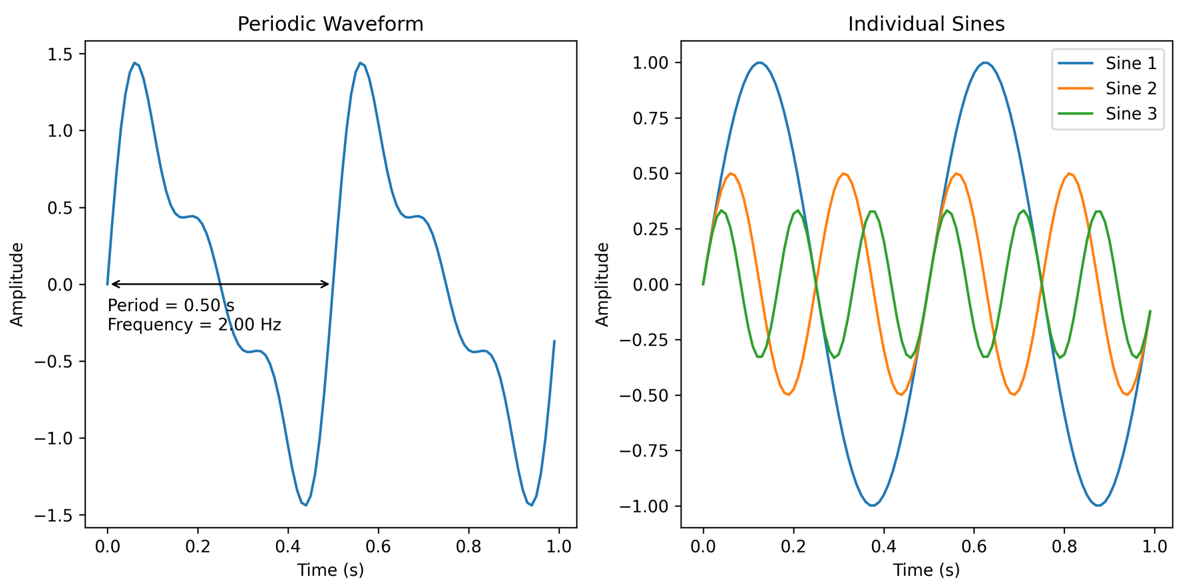
Sound phenomena can be described as variations in air pressure over time. We typically capture these variations using transducers called microphones that convert air pressure variations into voltage variations.

Joseph Fourier, a French mathematician and physicist who lived from 1768 to 1830, first proposed that any **periodic signal** can be expressed as a sum of **pure tones**.

A periodic signal is any signal that has a constant and repeating pattern over time. The length of the pattern is called the period, and the number of times the pattern repeats over one second is the fundamental frequency of the signal. Periodic signals usually exhibit **harmonics**. Harmonics are other frequencies existing in a periodic signal that are multiples of the fundamental frequency.

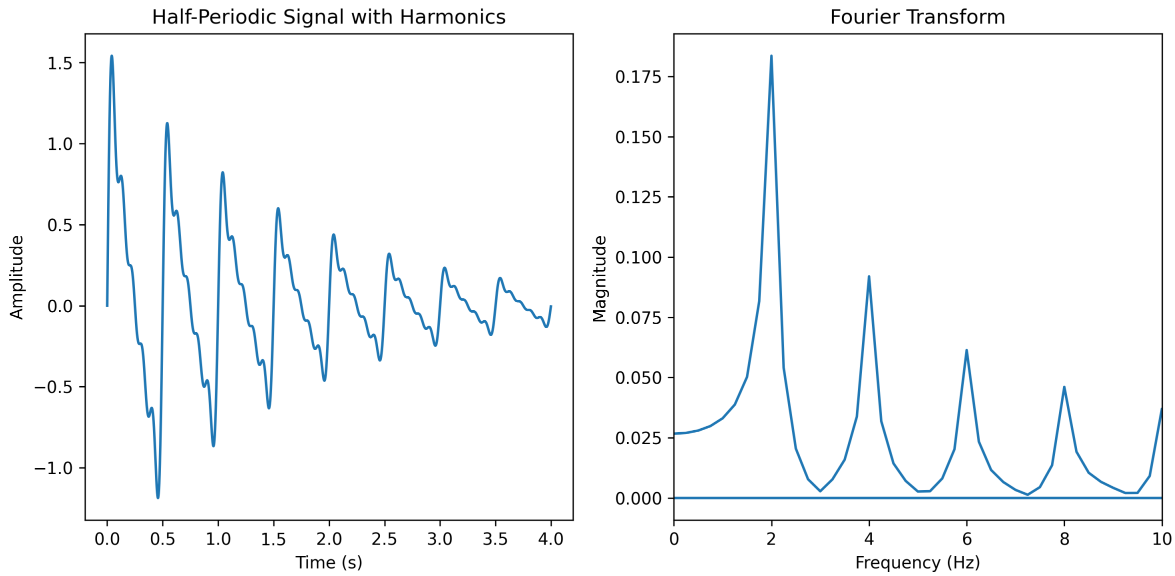


Pure tones correspond to particular periodic signals, which are also known as sinusoids. They have the unique property of having only a fundamental frequency and no harmonics. Due to this property, we can understand that a more complex periodic signal is the sum of multiple pure tones having frequencies and amplitudes that match those of the harmonics and the pure tone of the studied signal. This is the principle of the **Fourier Series**.



The Fourier series only applies to periodic signals. Most real-world signals, however, are not.

To overcome this problem, Fourier extended the Fourier Series to the **Fourier Transform**, which can be applied to any signal.



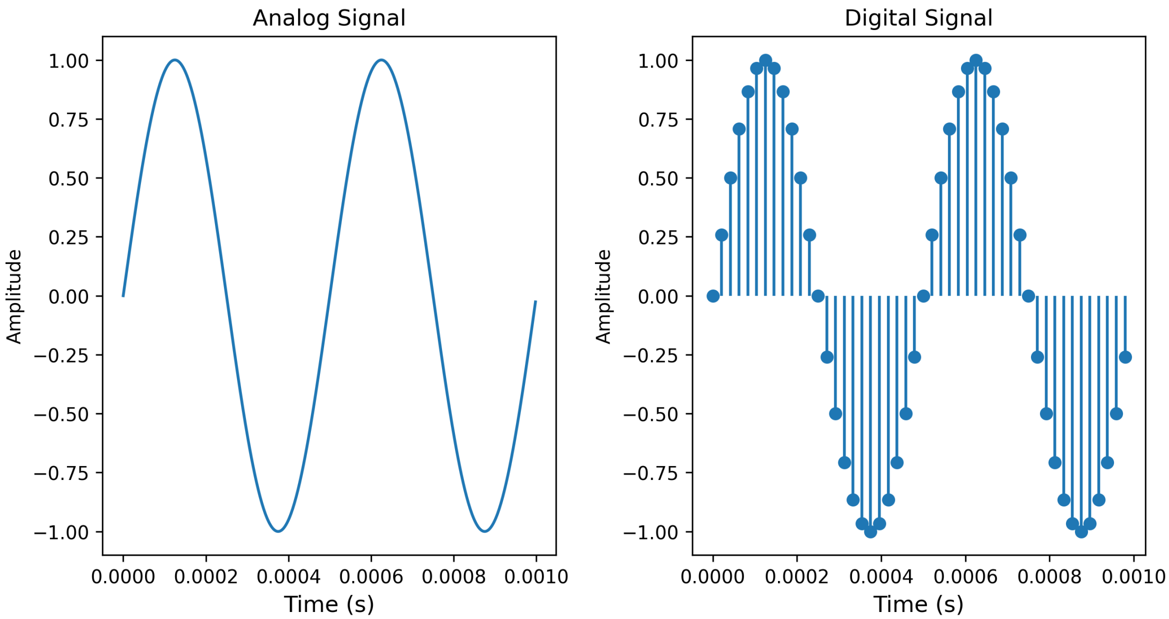
The Fourier Transform can be thought of as a band-pass filter bank where the output amplitude of each band-pass indicates the presence of sound energy in a certain frequency range.

The Fourier Transform is the foundation of many spectral analyses.

5.1.2 Fourier Transform in digital systems

So far, we have considered audio signals in the analog domain. The analog domain is characterized by time continuity, meaning that the value of an audio signal can be determined at any instant.

Computers cannot process analog signals because they require an unlimited amount of processing power and memory for any time span. To overcome this limitation, we sample analog signals, meaning that we take a value at regular intervals, transforming a continuous function into a discrete set of points. The frequency at which we take a value is called the **sampling rate,** and it also defines the maximum frequency that our digital system can correctly sample. According to the Shannon-Nyquist theorem, we must have a sampling rate at least twice as fast as the highest frequency in the signal we want to sample. Since it is generally accepted that human hearing does not detect sounds above 20 kHz, the typical sampling rate for audio signals is often around 40 kHz. This upper frequency limit is known as the **Nyquist Frequency**.



Since a digital system processes discrete audio signals, we must modify the Fourier Transform into a computable algorithm.

The definition of the Fourier Transform poses two significant limitations in the context of digital systems.

1. The Fourier Transform tries to identify any possible frequencies inside a signal. We need to define a limited range of frequencies to be able to perform this operation in a digital domain.
2. The Fourier transform also assumes that the signal is known throughout its lifetime. This restriction limits its use to offline analysis, making it impossible to apply in real-time.

The **Discrete Fourier Transform** was introduced to address the first problem. This method takes a completely known input signal and finds all frequencies with a period that is a multiple of the signal's length, up to the Nyquist frequency.

To address both problems, we use the **Short-Time Discrete Fourier Transform** (STDFT). This method allows us to analyze an incoming audio stream in several chunks, or buffers, rather than the whole audio file. We then follow the same process as for the discrete Fourier transform. Usually, the length of each buffer is referred to as the **window analysis length**. The STDFT outputs what we call frequency bins. Each frequency bin can be interpreted as a band-pass filter with a width equal to the inverse of the window analysis length.

The Fast Fourier Transform (FFT) is a specialized algorithm that we often use to calculate the Short-Time Discrete Fourier Transform. It has the particular property of being particularly efficient for buffer sizes that are powers of 2.

5.1.3 The uncertainty principle

When analyzing the content of an audio signal using an FFT, the length of the window analysis is a very important parameter to set up correctly.

A larger window increases the precision of our spectral analysis by reducing the step between each frequency searched in our audio signal. It also improves the low-end resolution in the context of audio signals. However, a larger window requires more samples from the input signal, making the analysis less responsive to rapid changes in the signal.

In simple terms, we can't have good frequency and time resolutions at the same time.

5.1.4 A first summarize

To get insight into a signal's spectrum, one needs to use the Fourier Transform. In the digital world, we use the Fast Fourier Transform, which is an optimized implementation of the Short-Time Fourier Transform. It can be performed in real-time. An FFT needs a window size, which corresponds to the number of samples taken into account for the spectrum analysis. A larger window size leads to more resolution in the low-end of the audio spectrum, while a shorter window size gives a better time reactivity. The FFT algorithm can have both a very good time and frequency resolution.

5.2 Understanding a Real-Time Spectrum Analyzer

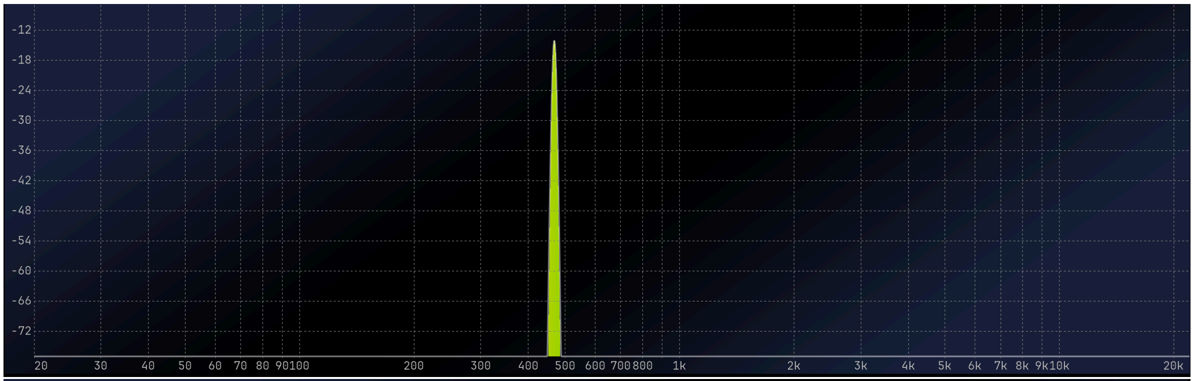
Real-time spectrum analyzers use FFT, or derivative strategies, to analyze incoming audio streams. Since such algorithms, as seen above, have limited frequency resolution, some inherent limitations are observed. Therefore, it is crucial for users to recognize and understand these constraints in order to accurately interpret the presented data.

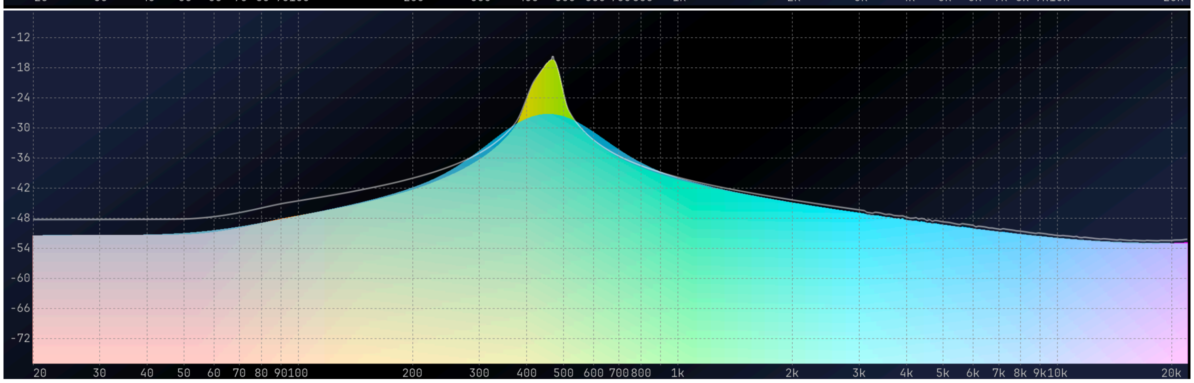
5.2.1 Why do some pure tones seem wider than others?

To demonstrate the MiRA real-time spectrum analyzer, we will conduct a straightforward use case by transmitting various pure tones and analyzing the results. To begin, let's adjust the main settings. The window size will be 1024, and the window shape will be rectangular.

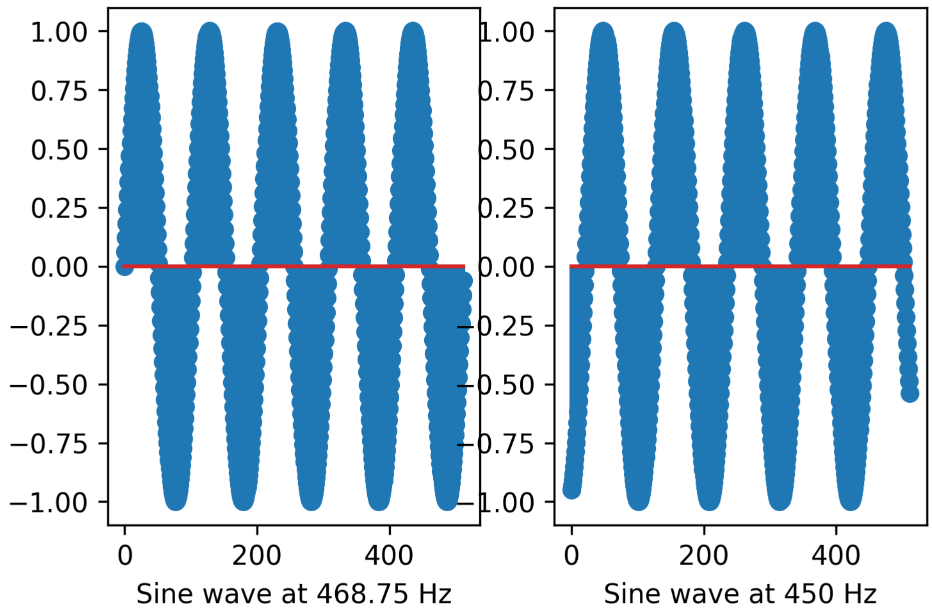
> **Caution**
>
> Unless you are absolutely certain of what you are doing, you should never change the default settings.

Now, we will set a sine wave generator to 468.75 Hz and observe the result. We observe a frequency spike that is quite narrow. However, if we slightly alter the frequency of our generator, we will observe a significantly different outcome.





At 450 Hz, the visualization generates a wide frequency peak with a high noise floor. Let's look at the two waveforms over a 1024-sample period to see what's happening.



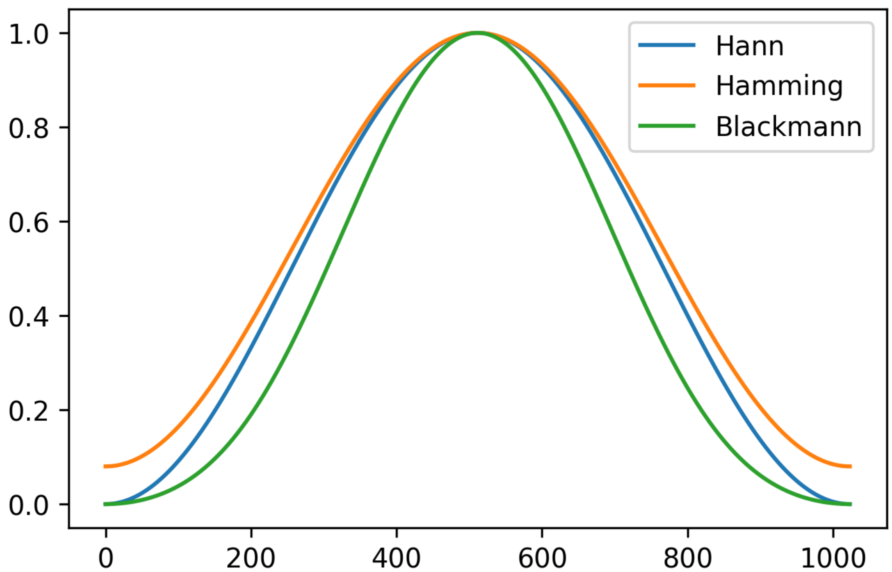
The left plot displays a waveform that precisely matches the length of the window size. In contrast, the right plot reveals a discontinuity. The FFT algorithm hears the discontinuity as a click, leading to the aforementioned issues in the resulting plot.

This phenomenon can be understood in a similar way to what occurs in an audio editor when attempting to edit without applying any fade-in or fade-out effects. You can hear clicks happening too!

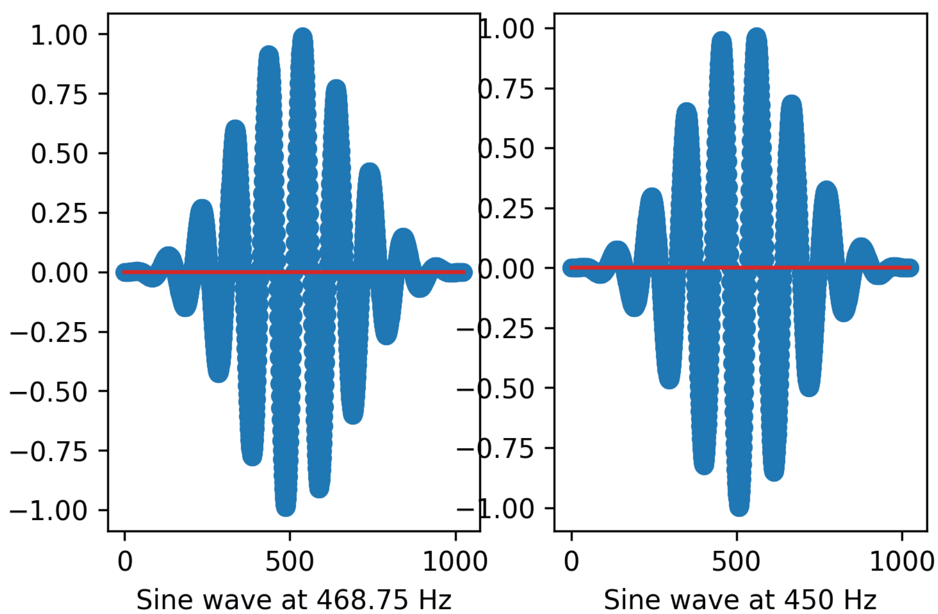
5.2.2 Windowing functions as fade in/fade-out

Windowing functions apply amplitude factors over a given time interval to smooth out discontinuities that may appear during the FFT computation.

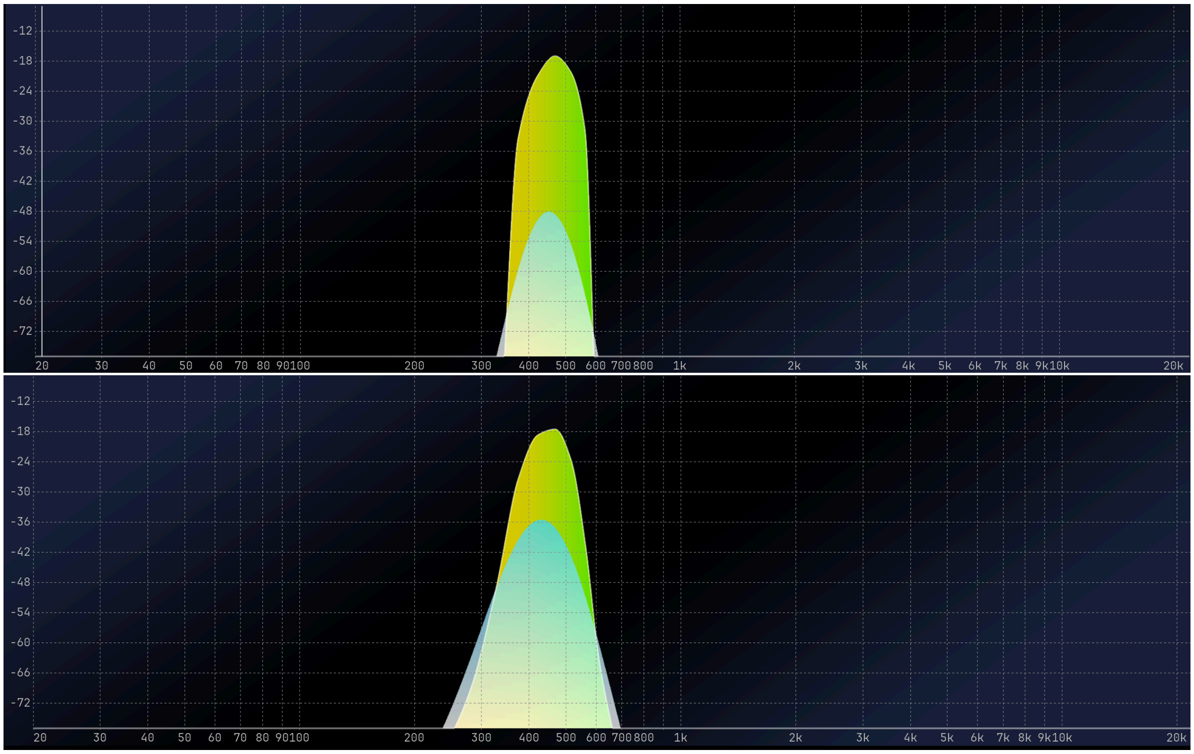
Here is a visual representation of various types of windowing functions:



The default window shape in MiRA is Blackmann. If we apply this windowing function to our two sine waves, here is what they look like:



Now, you can see that the buffer's extremity has faded to zero, thus removing the discontinuity.

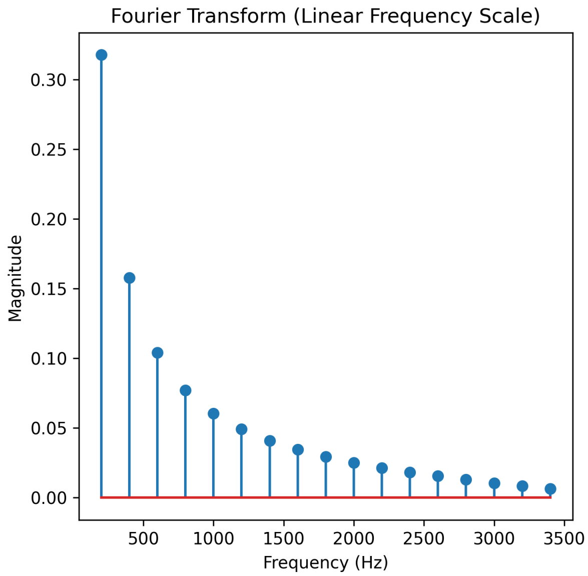


It's clear that our windowed sine wave at 450 Hz has significantly better results, although there is a slight decrease in accuracy for the 468.75 Hz sine wave. It is strongly advised that you consistently use a windowing function, as there is no justification for a signal to display frequencies that are only multiples of the FFT size.

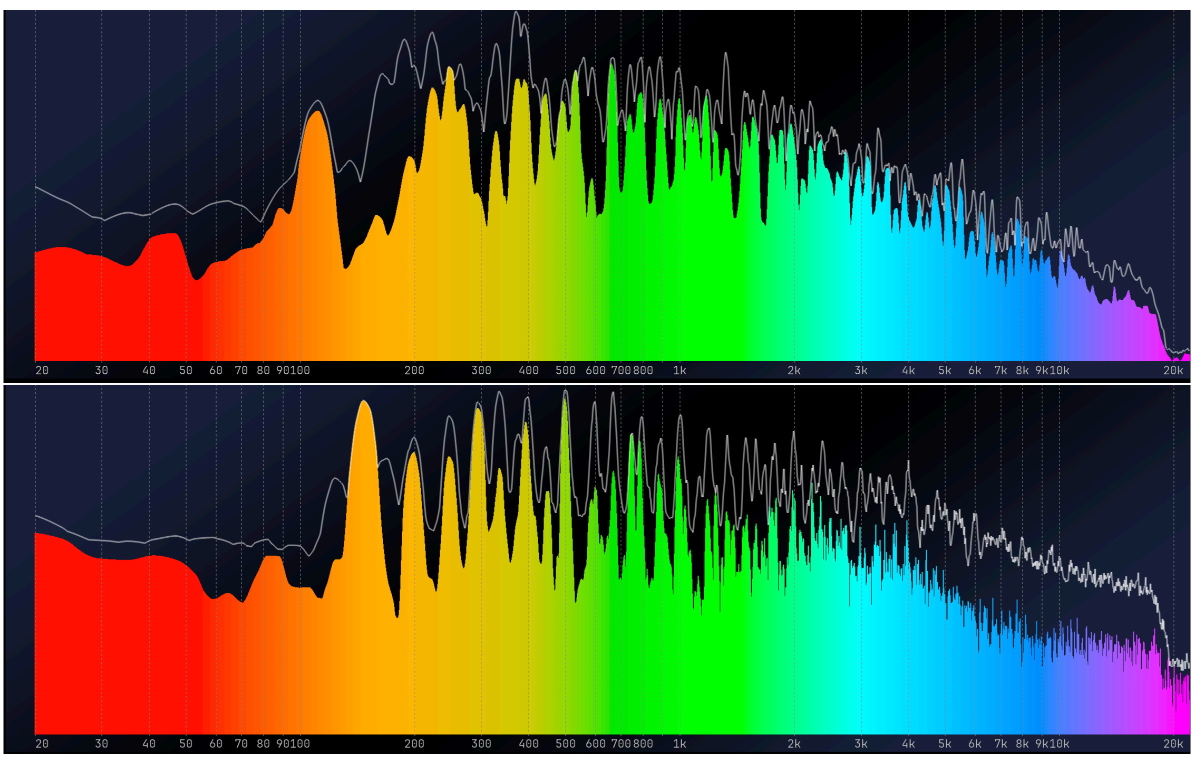
To achieve a higher degree of precision in frequency resolution, we can extend the analysis period. In MiRA, the default value of 8192 generally strikes a balance between frequency resolution and time accuracy.

5.2.3 Limitations of the Fast Fourier Transform

The Fast Fourier Transform is a valuable tool, but it has some limitations. One of the main ones is that it samples the frequency domain with a constant frequency step, which means that our logarithmic perception of frequency results in a higher resolution in the high-end of the spectrum and a lower resolution in the low-end. For example, the default window length of MiRA is 8192 samples, giving a frequency step of about 5 Hz. While this resolution is acceptable for the low end of the spectrum, it provides so much information on the high end that it makes it difficult to understand what is happening. Also, computing so many points requires a significant amount of CPU resources.



In the spectrum above, which represents a 200 Hz sawtooth, you can see that, when displayed logarithmically, its resolution is poorer in the low-frequency range. With richer signals, finer resolution in the high-frequency range can appear as noise. The measure itself is not noisy, but the greater point density can make plots challenging to read.



Ideally, we would like a Fourier Transform that corresponds to our logarithmic sound perception. Two implementations in MiRA:: can address this problem: the **Variable-Q transform** and the **Adaptive Resolution Transform**.

5.3 MiRA Propriatory Transforms

This section describes the two proprietary FLUX:: transforms: the variable-Q transform and ART. It details their advantages over plain FFT in the context of audio analysis, as well as their main use cases.

5.3.1 The Variable-Q Transform

The FLUX:: Variable-Q Transform (VQT) is an algorithm that matches auditory perception. As a reminder, our perception of frequency is logarithmic. For example, if we perceive a difference of one octave between two notes, the higher note has a frequency that is twice as high. The idea behind the VQT is to maintain a constant number of frequency bins per octave, thus maintaining a constant level of resolution with respect to our perception.

Such a strategy has several advantages:

- Although the variable-Q transform requires slightly more computing power than a simple FFT, it generates less data, making it much more efficient to use in MiRA.
- With its default window size of 8128 samples, the VQT provides a good resolution in both the low-end and high-end.

The Variable-Q Transform is the cornerstone of MiRA's real-time frequency analysis, since it is, by default, the engine of each scope displaying frequency information.

5.3.2 ART (Adaptative Resolution Transform)

The FLUX:: ART algorithm aims to solve two problems:

- Uneven resolution in the FFT when considering a logarithmic scale.
- The difficulty of obtaining good temporal and frequency resolutions.

The idea of ART is to use multiple FFTs of different sizes, depending on the octave under consideration. Higher octaves can use very short FFT sizes, which gives them excellent time resolution. At the same time, this keeps a good frequency resolution for the part of the spectrum we are considering. For lower octaves, however, the FFT size increases in order to maintain good frequency resolution.

When combined, all of this data provides a fairly consistent distribution of points, taking into account our logarithmic perception, while remaining quite sensitive to very fast information such as transient.

Thus, ART is the default algorithm for our transfer function scope.

Part V

User Interface

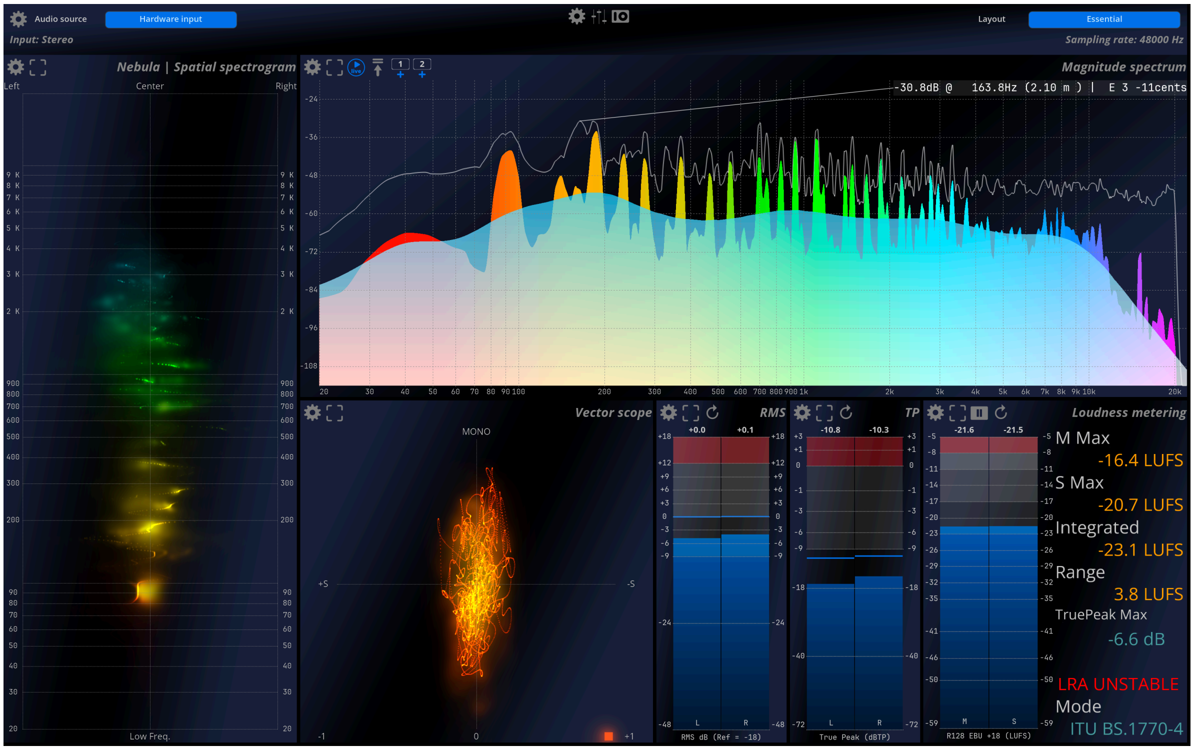


Figure 5.1: Default view of MiRA

The FLUX:: MiRA:: has a modular interface by design. Each audio analysis tool is called a **scope**. An ensemble of scopes is called a **layout** and it is the main display of the app. Layouts are designed to regroup scopes that make sense for specific usage and use cases. Layouts are user-manageable: you can create new ones, delete existing ones and arrange the scopes inside of one exactly as you wish.

App structure and lexicon

Having a clear understanding of the MiRA:: structure is of a major importance to use it efficiently. First we need to address a few terminology:

- A **scope** is an audio analysis tool, like a real-time spectrum analyzer, or a loudness meter. We can then say that each part of the GUI that displays some kind of information is a **scope**.
- A **layout** is an organization of **scopes** and of **containers** on the screen. The *MiRA::* is shipped with several default layouts that should handle most of the major use cases, but users can also **create their own custom layouts**. Containers are building blocks of the layouts. You can think of them as a subset of scopes, and they are useful to create

more complex layout. A layout also can also recalls specific configurations presets (for UI, IO and Main preferences).

- A **workspace** is an ensemble of layouts. It also **stores all the app's global preferences** (IO/UI/Settings menu). Workspaces are user manageable and can be saved, opened, duplicated, etc.

We can now visualize these different elements as boxes in boxes:

- A scope is an audio analysis or a user interface tool
- A container holds scopes
- A layout holds scopes and containers
- A workspace holds layouts and global app settings.

6 Layouts & Workspaces

A layout is a collection of scopes that are designed to perform a specific task. A scope is a visualization tool, such as a spectrum analyzer or a loudness meter. A layout can also recall specific presets for UI, IO and main setup.

For example, studio-oriented layouts usually focus on loudness and real-time spectrum displays, while live-oriented layouts favor Leq and transfer function scopes.

The MiRA:: allows you to fully customize all the different layouts, even creating new ones or deleting existing ones from the factory template.

6.1 Factory layouts

These default layouts are designed for specific use cases and can serve as a good starting point. The number and nature of these layouts vary based on the version of MiRA:: you use.

Most of these layouts include the Info Header Bar, which provides quick access to the main, I/O, and UI settings menus, as well as some important settings. See info header section 9 . Those that do not use it are included to demonstrate that the layouts are fully customizable. You can exit these layouts by using the `TAB` or `Shift+TAB` shortcuts, or the top menu `View > Layout`.

6.2 Workspace

A **workspace** consists of a set of layouts that you can navigate in the MiRA app. It also stores all the app settings (main, UI and IO menus). In practice, you can create multiple workspaces for different use cases. For example, you could have a workspace for studio use, with specific layouts and I/O configurations, and another workspace for live performance, with different layouts and dedicated I/O configurations.

6.2.1 Basics

When MiRA:: starts, it automatically loads the factory default layouts. You can navigate through the different layouts using the `TAB` and `Shift+TAB` shortcut, or by selecting the `View>Layouts` menu item from the top application.

Another solution is to display the layout bar. To do this, go to `Edit>Show Layout bar` in the top menu, or use the `CMD/CTRL+L` keyboard shortcut. This layout bar allows you to modify the layout using the central drop-down menu. For more details on how to create and customize a layout, see the section @seq-customeLayoutEditor.

To save a workspace, go to `File>Save Workspace` or `File>Save As` in the top bar. The workspace data is stored in a '.json' file.

You can reopen a previously created file using the `File>Open workspace` option in the top menu.

> Caution
>
> .json files are used by many software to store various types of data. Make sure that you are trying to open a MiRA:: workspace.

6.2.2 New workspace

To create a new workspace, execute the `File>New Workspace` action. As there is no layout in this workspace, MiRA:: will simply display an empty black screen. You will have to create your own custom layout to populate the screen with some scope.

If you prefer to start your custom workspace from the factory layouts, you can use the `File>New workspace from factory template` action.

6.2.3 Canceling unsaved modifications

If you want to undo the modifications that you have made to a workspace, you can do so by reloading it as long as you have not saved the changes.

To reload a workspace, use the `File>Reload workspace` action.

7 Workspace Bar & Layout Editor

MiRA:: allows you to create custom layouts. A layout is an ensemble of scopes that can be placed anywhere on the user interface. It also stores the scope settings. The layout combines a grid system and a container system, allowing the user to create any possible layout.

7.1 Workspace Bar

To open the workspace bar, use the shortcut `CMD/CTRL+L`, or navigate to the `Edit>Show Workspace toolbar`

Figure 7.1: The workspace bar

The workspace bar gives access to the organization and modification of the layout present in the current workspace. The current displayed layout can be changed using the drop-down menu and selecting the desired one. The selected layout can also be moved to the list position using the two arrows on the right of the drop-down menu.

The **Edit** button activates the editor to customize the currently selected layout. The **Advanced** button toggles on and off the advanced editor.

The **New** button creates a new blank layout.

The **Rename** button renames the currently selected button.

A layout can be duplicated using the **Duplicated** button, or deleted using the **Delete** button.

On the right of the workspace bar, you will find a drop-down menu to attach specific **UI**, **IO** or **main** menu setup **presets** to the currently selected layout.

7.2 Layout Editor

To edit the currently displayed layout, use the shortcut `CMD/CTRL+SHIFT+L`, or navigate to the `Edit>Show Workspace toolbar` menu and to `Edit>Layout>Edit current`.

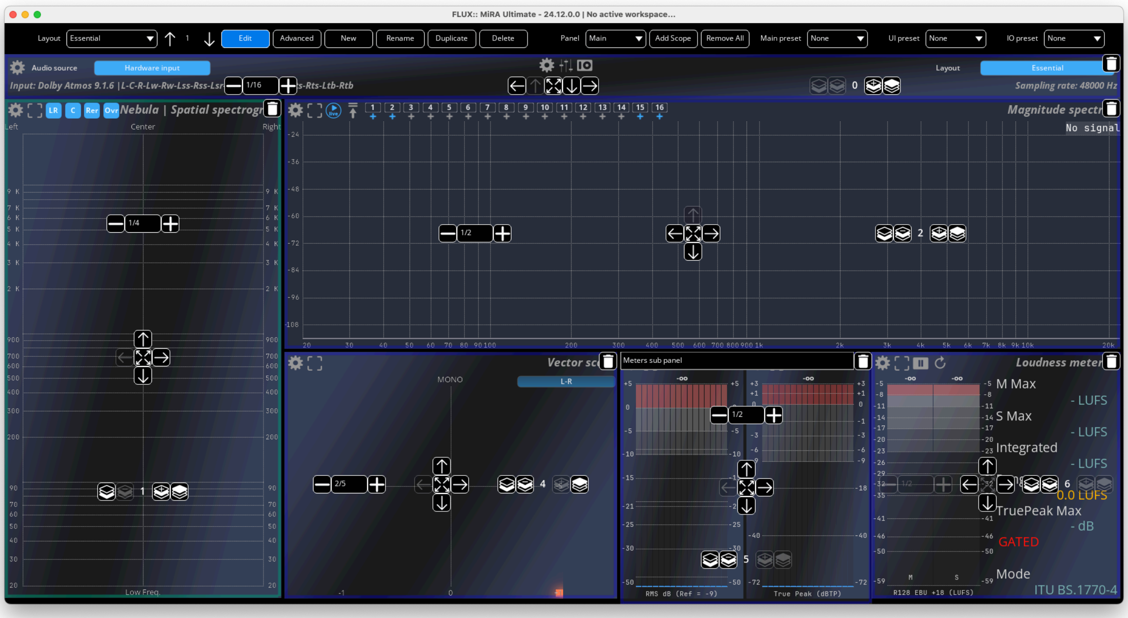
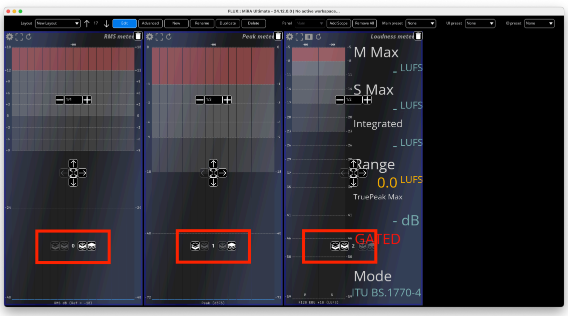


Figure 7.2: Layout being edited

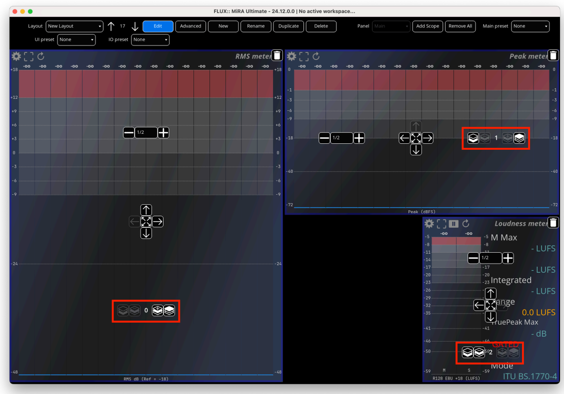
Several buttons appear on each scope:

- The arrows in the middle of the scope allow us to choose an anchor edge. When using the center button representing four arrows, the scope will occupy all the available empty space.
- The drop-down on the left allows us to change the dimension of the scope. If the scope is attached to the left or right sides, it sets the width of the scope. If the scope is attached to the top or bottom sides, it sets its height. If the scope occupies all the available space, then the drop-down is grayed out. The two little `+` and `-` buttons allow for fine adjustment over the size of the scope.
- The right side feature controls to change the z-coordinate of the scope. The editor uses an "above-bellow" logic to place the scope on the user interface. This setting has two main influences:
  - The size of the scope is computed relatively to the z-coordinate of the scope. The scope at the lowest position has its size computed against the whole user interface.

The scope at a given level has its size computed against what is left unoccupied in the user interface.
  - When several scopes are attached to the same edge, the bottommost scope is the closest to the edge.
- A recycle bin icon on the top right corner allows for scope deletion.



(a) Scopes are all attached to the same edges



(a) Scopes are attached to different edges

7.3 Creating A New Layout

To create a new layout, we need to open the workspace bar (`CMD/CTRL+L`) and the click on the `New` button. This will create a blank layout that then needs to be populated with scopes.

To add a scope, click on `Add Scope`, then choose the desired one. As seen before, we can now access the position options in this new scope. It can be:

- Attached to the top, the left, the right, or the bottom of the layout.
- Moved in the z-axis.
- Deleted.

Usually, we choose to which side of the layout we want to attach the scope first, then, we use the size button to change the size of the scope. Once you are satisfied, click on `Add scope` again to add another scope and repeat the same step to position it on the screen. If you need to alter the scopes' hierarchy, simply change their z-coordinate.

Finally, exit the Edit mode by clicking on the `Edit` button.

7.4 More complex layouts with containers

Some specific scope positioning is impossible using only what we have covered so far. For example, you cannot place a scope in the corner of the screen. We need to use containers to achieve a more specific and complex layout.

To add a new container, click on the `Add Scope` button and choose `Scope Container`. Now, each new scope will be added to the container.

> **! Important**
>
> Note that nested containers are not allowed.

You can navigate between containers and the main layout using the container selector drop-down menu is the workspace bar. Also note that, from the main panel, the containers can be positioned in the user interface just like any other scope.

As an example, if you want to position a vector scope in the top-left corner, start by creating a container. Attach it to the left side of the screen. Then select the container from the drop-down menu, add a new vector scope, and attach it to the top.

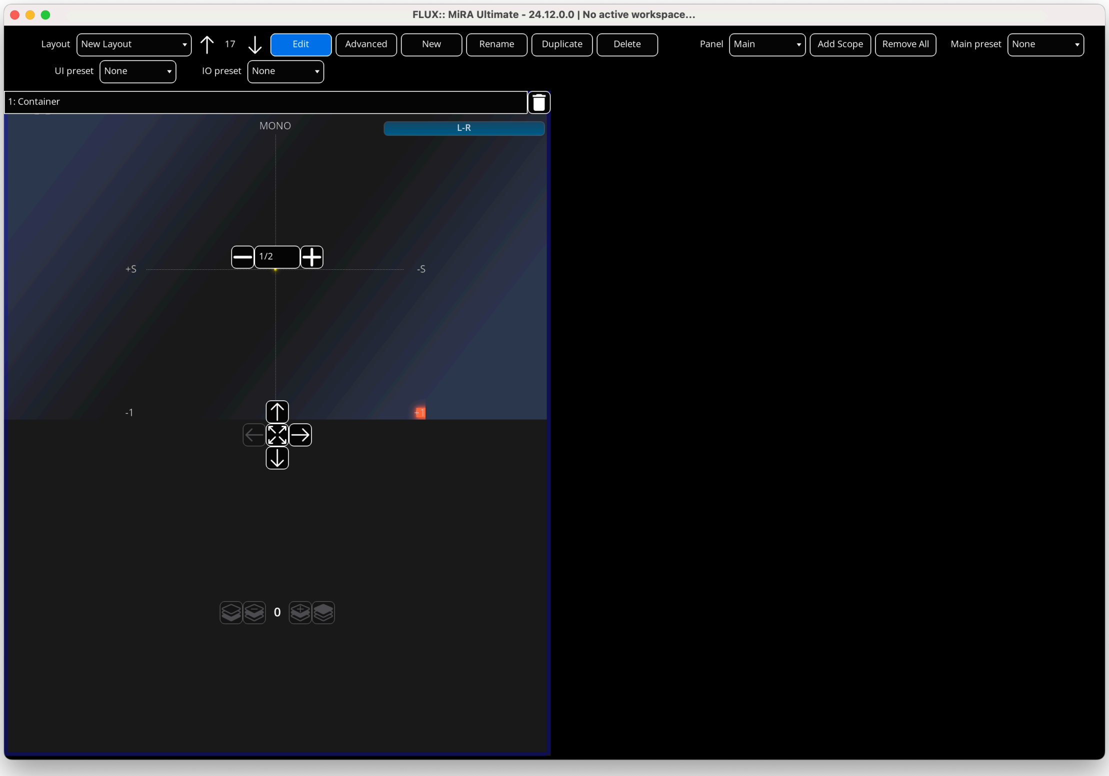
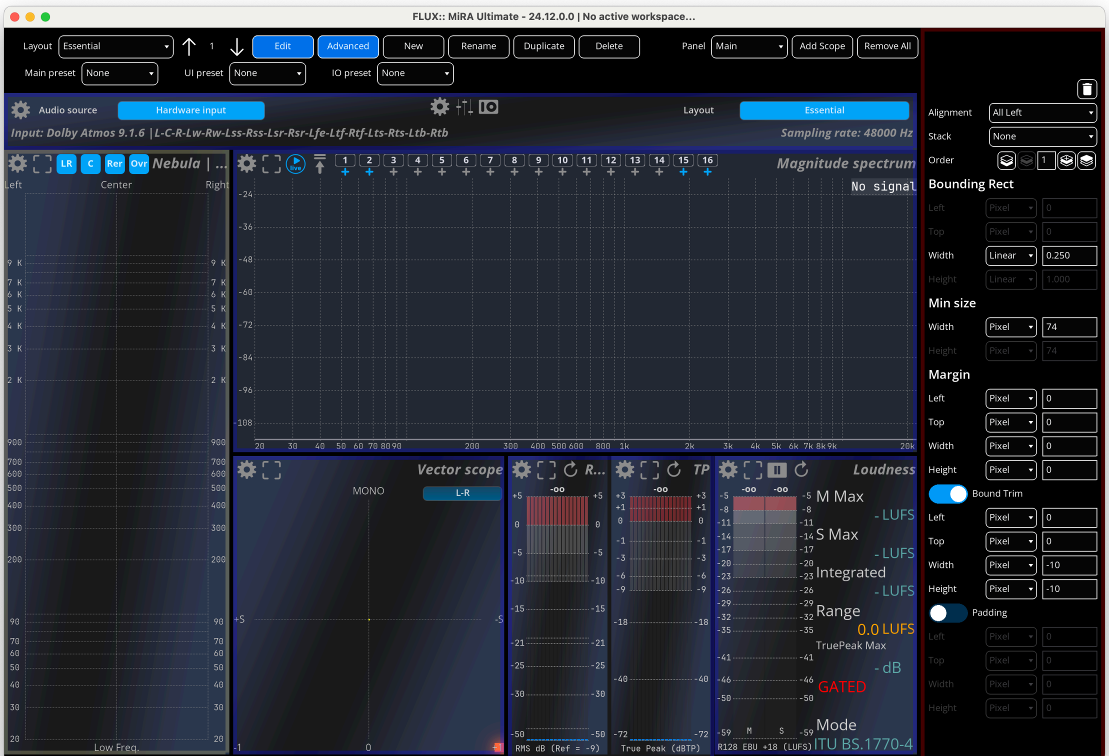


Figure 7.5: Vector scope in the top left corner

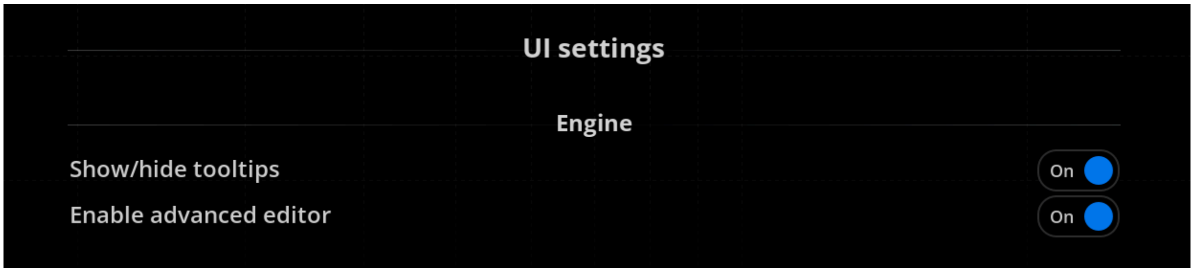
8 Advanced Layout Design



MiRA has an advanced layout design mode that enhances the customization options for your layouts. This mode provides additional features and controls to fine-tune your layouts, making them more versatile and tailored to your needs.

8.1 Activating Advanced Mode

To activate the advanced layout design mode, first reach the UI menu and activate the option labeled “Enable advanced editor”.



Then, a new button appear in the workspace, named "Advanced", which allows to toggle between the classic and the advanced editor.

8.2 Features of Advanced Mode

8.2.1 Alignment

In advanced mode, you have more options for automatically aligning controls. The five buttons with arrows correspond to "All", "All Left", "All Right", "All Top", and "All Bottom".

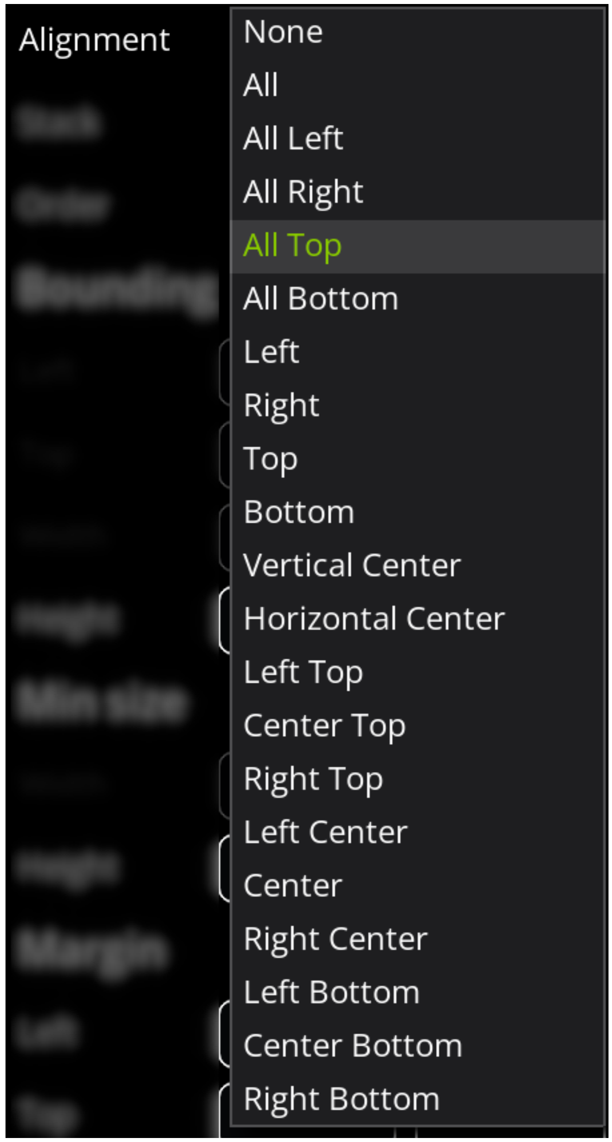


Figure 8.1: Alignments

8.2.2 Stack

Controls can now be stacked vertically, horizontally, or both (horizontally first).

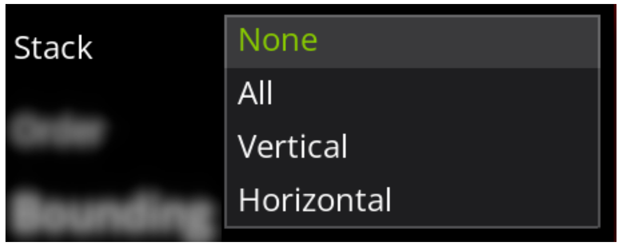


Figure 8.2: Stacks

8.2.3 Metrics Options

Each metric can now be set in “Pixel” (fixed size), unlike the classic mode which only allows “Linear” metrics (relative to the container). Additional options include:

- **Height/Width Factor**: A linear factor of the other coordinate (e.g., “Height factor” can be applied to horizontal values and vice versa).
- **Min/Max Factor**: Applies a linear factor from the minimum or maximum value of the height and width of the scope.

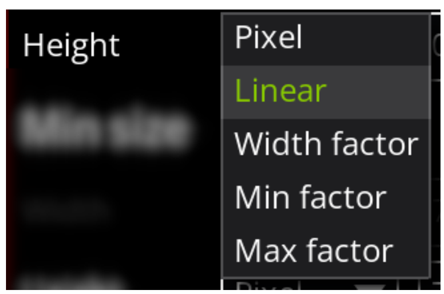


Figure 8.3: Metrics Options

8.2.4 Min Size

Set the minimum size of the control to ensure proper display, especially when the scope size is relative to the window.

8.2.5 Margin

Add space around the selected scope by setting a margin.

8.2.6 Bound Trim

Unlike margin, bound trim values do not depend on alignment and surrounding controls.

8.2.7 Padding

Increase the size of the selected scope while keeping the content the same.

9 Info Header Bar

The Info Header Bar, which can be found in most factory layouts, provides quick access to various menus and settings. It is divided into three parts. On the left, the Audio Source Toolbar allows you to change the input source from hardware to a detected SampleGrabber instance on the network and displays the current input configuration. In the middle, the Quick Access Toolbar has three buttons: the cogwheel for main settings, the UI button for user interface settings, and the IO button for input/output settings. To the right, the Layout and Samplerate section includes a drop-down menu for selecting a workspace layout and displaying the current sample rate. Also note that the Info Header Bar does not appear by default in new layouts, but must be manually added to the user interface.

9.1 Audio source toolbar

On the foremost left side, we can find:

- A drop-down menu to switch between hardware and detected sample push instances on the network.
- A display of the current input config.

9.2 Quick access toolbar

In the center, there are three buttons:

- The cog, which is the main settings menu.
- The UI button, which opens the user interface settings menu.
- The IO button, which opens the input/output settings menu.

9.3 Layout and sample rate

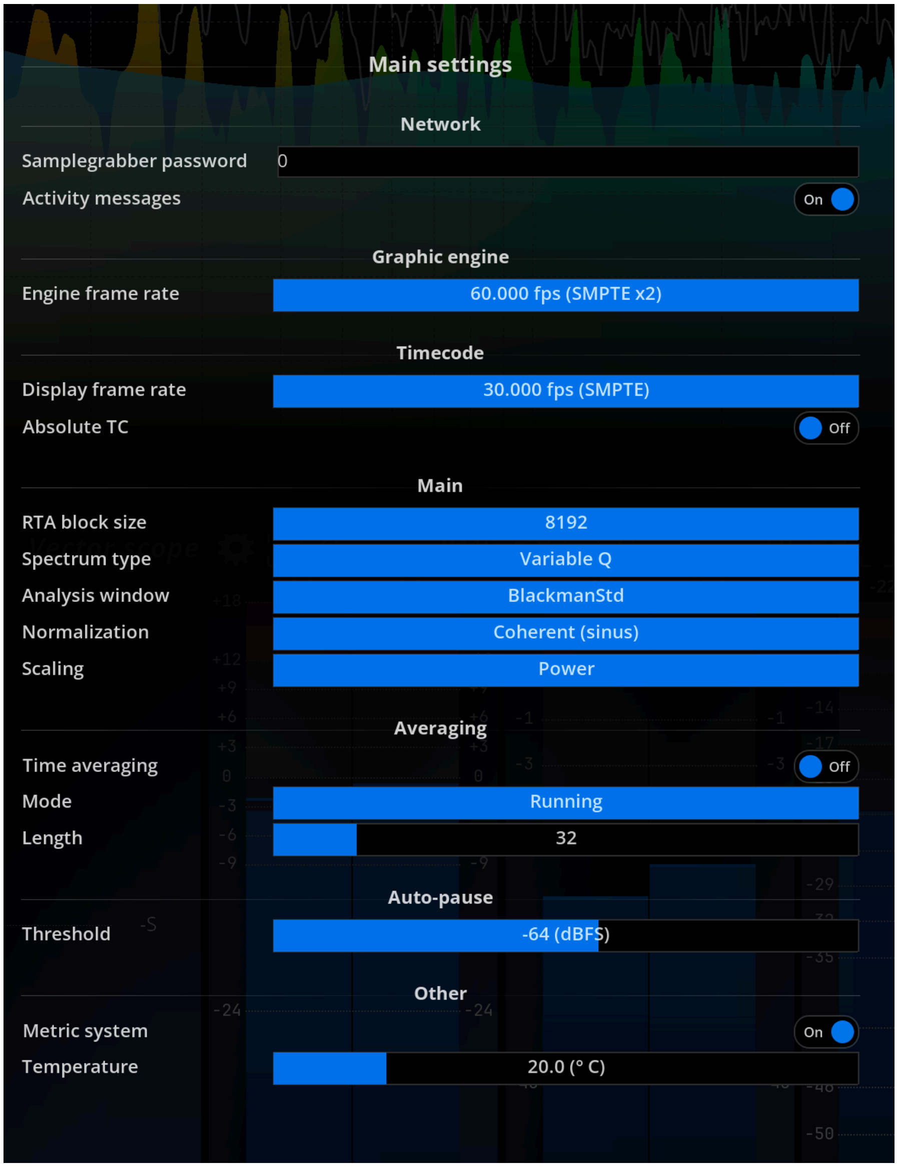
Last, we find, at the foremost right:

- A drop-down menu to select a layout of the workspace
- The sample rate display.

> ❗ Important
>
> The info bar header is not present by default inside a new layout. You should manually add it as part of your user interface.

10 Main Setup

The Main Setup section allows you to configure various aspects of the application, including saving and restoring user-defined configurations, setting up the SampleGrabber password for secure audio material access, specifying graphic engine frame rates, adjusting timecode settings, and configuring main analysis parameters such as RTA block size, spectrum type, TF/Sweep block size, overlap mode, analysis window, normalization, scaling, and averaging. Additionally, you can set various preferences, like the auto-pause threshold, metric system, temperature, and reset preferences to their default values. Each setting is designed to optimize the performance and usability of the application based on your specific needs.



Main setup dialog

10.1 Configuration

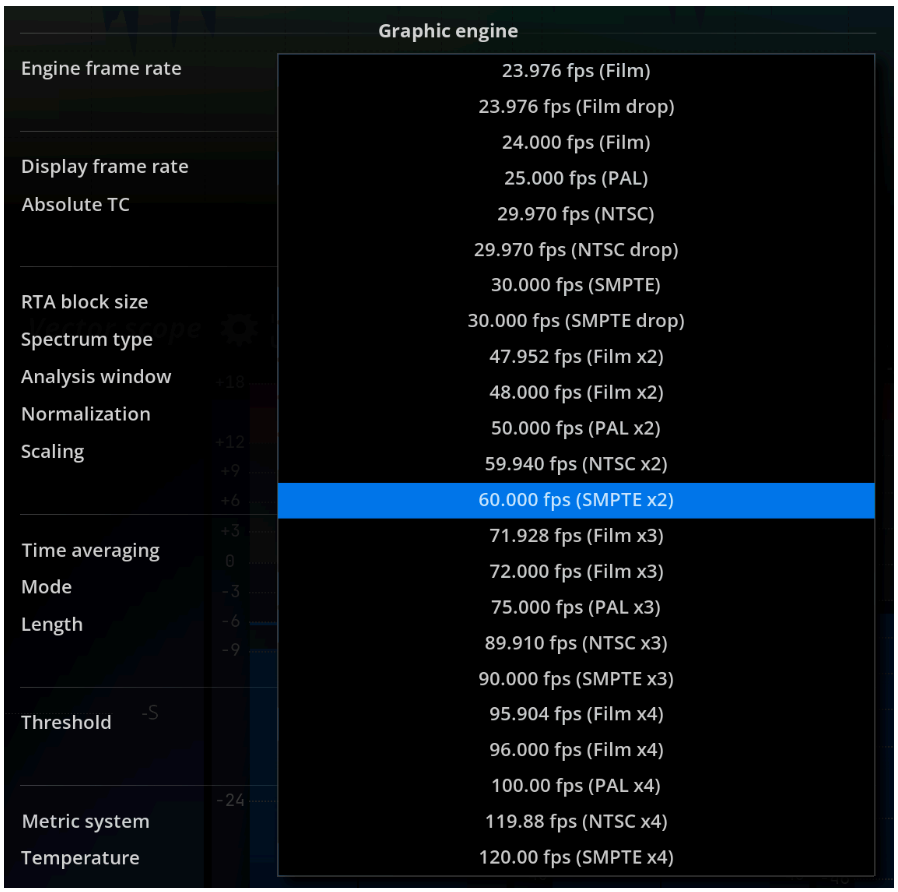
Save/restore a user-defined configuration to and from disk, including all the settings in this panel, as well as IO Configuration 12 and UI Setup 11.

10.2 SampleGrabber

SampleGrabber password

The password entered in this field should match the one used by the SampleGrabber you wish to use as a source. This provides a reasonable level of security and prevents unauthorized access to your audio material broadcast over the network. Please consider that the encryption used only provides moderate protection, and is not intended to replace other security guards, such as firewalls, etc.

10.3 Graphic engine



Note

Available graphic engine frame rates

Here, you can specify the rate at which the display should be refreshed. Please note higher frame rates place higher demands on the GPU, and, to a lesser extent, on the CPU.

> The effective frame rate can be displayed by typing `SetRenderStats(1)` in the console.

10.4 Timecode

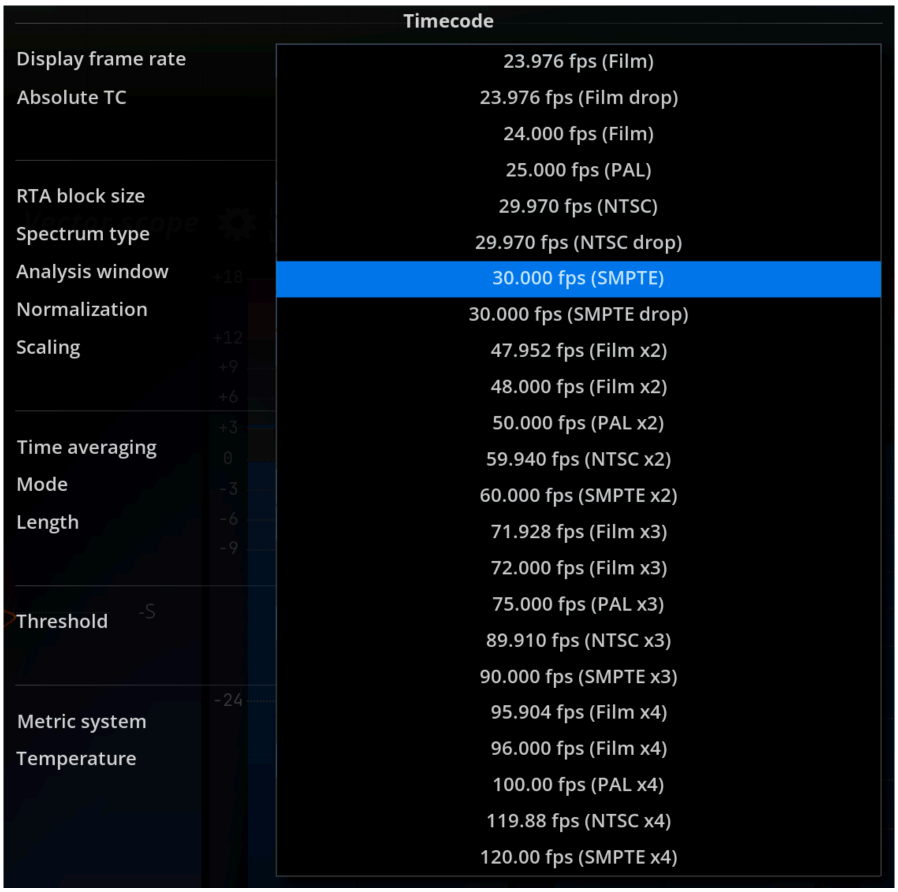


Figure 10.1: Available display frame rates

10.4.1 Display frame rate

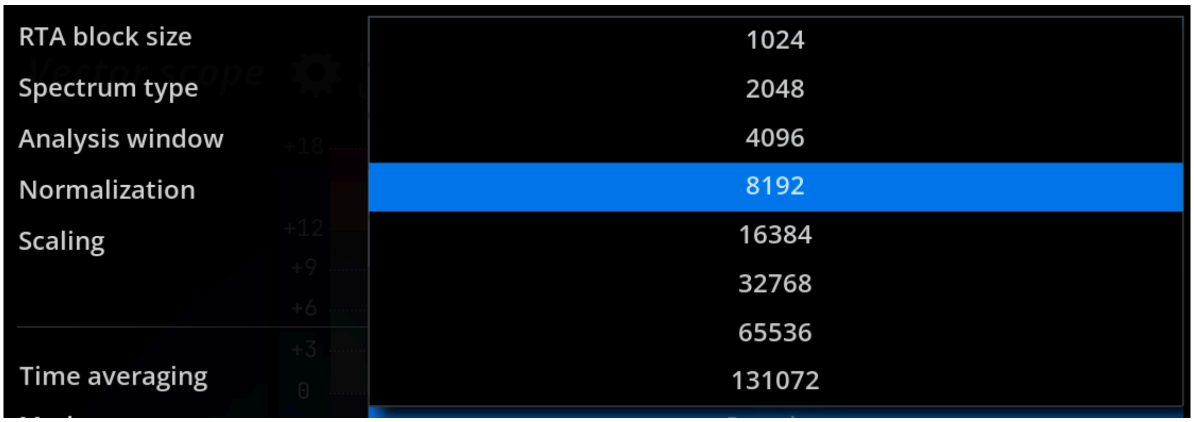
Sets the frame rate used for time display in various parts of the program. Set it to match the frame rate of your source material to facilitate locating time events when working with film, TV or other time-stamped material.

10.4.2 Absolute Timecode

This setting toggles between absolute and relative time-code display formats. Absolute Timecode is taken from the time the application was started. Relative Timecode is the time-elapsed since the Timecode offset position. See metering history usage 25.1 for information on working with Timecode.

10.5 Main

10.5.1 RTA block size



Defines the size of the blocks in samples, fed to the main spectrum analyzer engine, which is used by the spectrum magnitude, Nebula and spectrogram views.

Keep in mind that the incoming audio needs to be accumulated in a buffer for a certain amount of time before the data can be computed and the display updated. In contrast with the buffers you probably know from sound cards, this block processing is not just a computer technicality nor a source of undesirable latency, but an integral part of the analysis process.

As such, it determines both the precision of the analysis and the maximum display rate, and should be adjusted depending on the specifics of your application.

> **Note**
>
> In order to maintain a sufficiently responsive display refresh rate, blocks overlap by 75 %.
> The default setting is 8192 samples, corresponding to a length of roughly 180ms at 44.1kHz sampling rate. This value constitutes a good compromise between precision and responsiveness for most situations. However, if you need to measure a particular frequency with great precision, you should raise the analysis block size. On the other

hand, if you need to follow rapid spectrum variations, this value should be lowered.

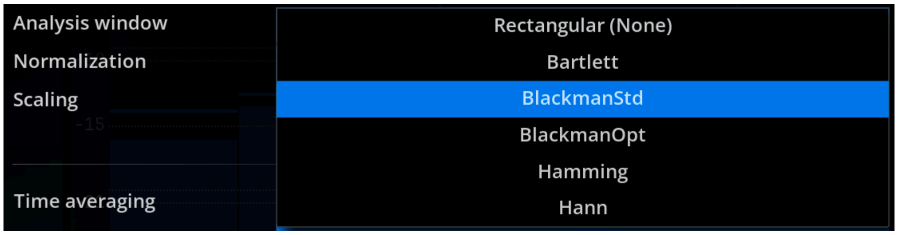
10.5.2 Spectrum type

Toggles between Variable Q mode (default) and standard FFT.

The discrete Fourier transform (DFT) is the traditional method employed to compute the frequency spectrum of a discrete digital signal, which is implemented using the FFT algorithm in MiRA.

FLUX:: MiRA:: Variable-Q uses both standard DFT and proprietary algorithms that more closely model human perception.

10.5.3 Analysis window



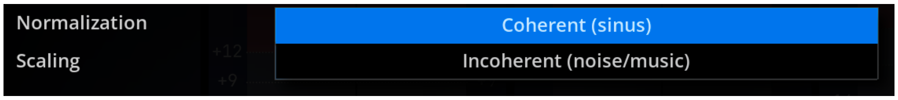
The first step of signal analysis is to split the incoming signal into overlapping blocks. Each block is then multiplied with a so-called window signal prior to the spectrum computation. The purpose of this is to minimize the side effects of block processing, such as the introduction of transients at the block boundaries, etc.

Available choices are:

- Rectangular (None).
- Bartlett.
- Blackmann standard (default).
- Blackmann optimized.
- Hamming.
- Hann.

We suggest you leave this setting to the default unless you are quite knowledgeable about these aspects, or if you should need to explicitly recreate a specific measurement, such as a particular method specified in a standard document.

10.5.4 Normalization



Selects the normalization mode used to normalize the global gain of the spectrum display.

Available choices are:

- Coherent (sinus): 0dB peak sine gives 0dB amplitude.
- Incoherent (noise/music): 0dB RMS noise or music gives 0dB power.

10.5.5 Scaling

This setting controls the frequency dependent amplitude spectrum correction curve.

This affects how various standard reference signals register on the display. The default *power* scaling will result in a signal with spectrum components of *constant power* registering as a flat curve, whilst amplitude will have the same effect for components of constant *amplitude* such as pure tones (sine signal).

The table below shows how the curve appears depending on the type of input signal. 1/f corresponds to a rectilinear slope on the display with both X and Y axis being logarithmic.

Input signal	Sine	White	Pink noise
Power scaling	1/f	1/f	Flat
Amplitude scaling	Flat	Flat	1/f

For monitoring a mix, it makes the most sense to use *power* scaling, as this is the way our hearing responds. If you need to measure a room's acoustic response, an outboard unit or a plugin's frequency response, the system's magnitude transfer function is best suited for this purpose and scaling has no effect.

The *amplitude* scaling setting should therefore really be employed if you need to measure relative amplitude values, such as those of sine test tones at various frequencies. Also, note that plain DFT corresponds to scaling set to *amplitude*.

The power of a time signal is proportional to the square of its amplitude, or equivalently, its power in dB is double the amplitude. However, in the case of a spectrum, we are measuring the output of a filter bank, which reacts very much differently depending on the type of input signal, so the simple previous formula doesn't apply anymore.

Available choices are:

- Amplitude: equivalent to no scaling. The amplitude of pure tones at different frequencies registers at the same value. Incoming white noise is displayed as a (quasi) flat curve.
- Power (default): scaling inversely proportional to frequency (1/f). Incoming pink noise is displayed as a flat curve.

10.6 Averaging

Time averaging: engages averaging of spectrum magnitudes over time. Default is off.

10.6.1 Mode

- Running: the average display is updated as soon as a new incoming block arrives. This is the default.
- Fill-freeze: the display is only updated when a fresh batch of N new incoming blocks has arrived. The display is frozen until the next batch of N blocks arrives, and so on. N corresponds to the length setting defined below.

10.6.2 Length

The number of incoming blocks over which the resulting average spectrum is computed. Lower values lead to faster apparent display update rates, while higher values smooth-out any time-variations more. Default is 32.

> **Note**
>
> Running average employs a weighting window that gives more importance to the last incoming blocks of samples. This type of time averaging is also called moving average, rolling average or running average, and is good for smoothing out abrupt variations in time and still be able to monitor in a continuous fashion.
> Fill-freeze mode is useful for stabilizing a flickering display while still following long-term variations, which permits a more detailed study of the curve(s). This mode is therefore

useful to get a very steady picture of the spectrum while still monitoring some of the mid-term changes, and saves you from holding and resetting the display manually again and again.

10.7 Various

10.7.1 Auto-pause threshold

Analysis is paused whenever the level of any channel of the incoming audio falls below this level. Set this a tad above the acoustic and electronic noise floor of your input signal chain to retain measurements even though the audio (music program or test signal) has stopped.

10.7.2 Metric system

Toggle displayed units between:

- Metric system (default): distance expressed in meters, temperature in degrees Celsius.
- Imperial units: distance expressed in inches and feet, temperature in degrees Fahrenheit.

10.7.3 Temperature

This should be set to the ambient temperature at the current location in order to get the most accurate time to distance conversions in the delay finder and impulse response panels. The following table gives an idea of how much the speed of sound varies with temperature.

Temperature (°C)	Speed of sound (m/s)
0	331.3
15	340.31
25	346.18
35	351.96

10.7.4 Preferences reset

Resets "Default" application configuration settings to their default initial value. Please note the changes are only effective after restarting the application.

11 UI Setup

The UI Setup guide allows you to customize your user interface preferences, fonts, and colors to enhance your experience. In the Preferences section, you can save or restore your complete user-defined configuration. The “Fonts” section lets you adjust the size of various fonts used in the interface, such as grid labels and spectrum peak labels. Additionally, the Colors section provides options to adjust the global brightness and contrast, and even switch to a reverse color scheme for better readability in outdoor environments.

11.1 Preferences

Name	Description
Configuration	Saves/restores a complete user-defined configuration.

Figure 11.1: User interface setup dialog

11.2 Engine

Name	Description
Vertical sync	Prevent the appearance of screen tearing by synchronizing the application's framerate with that of the screen.

11.3 Fonts

Name	Description
Small Scale	Sets the size of the smallest font used for drawing the grid labels.
Large Scale	Sets the size of the largest font used for drawing the grid labels.
Spectrum Peak Label	Sets the size of the font used for the Spectrum peak label.

11.4 Colors

Name	Description
Brightness	Adjusts global user interface brightness.
Contrast	Adjusts global user interface contrast.
Accentuation	Define the accentuation color of MiRA
Background	Define the background color of the app
Scope Solid 1	Define the first scope solid color
Scope Solid 2	Define the second scope solid color
Gradient color 1	Define the first scope gradient color
Gradient color 2	Define the second scope gradient color
Gradient color 3	Define the third scope gradient color

12 IO Configuration

Figure 12.1: IO configuration dialog

The MiRA:: app supports up to 24 input channels. There are two main usages for this audio stream:

- Feed the RTA system; also called *input reference* (red parameters & information)
- Feed the capture system; also called *TF input* (green parameters & information)

> ❗ Important
>
> The capture system is only available in MiRA::Live version

Although the audio channels are shared by both, it is of major importance to make sure that both systems are **independent**. Thus, one IO modification made on one does not affect the other one (as long as it is not common setting, such as the choice of audio interface, sample rate, etc.).

12.1 Source

12.1.1 Audio source

Audio source allows you to select which source to use as input. Depending on your current configuration and settings, this will include:

- Available SampleGrabber instance(s), either local or remote.
- Available hardware device(s), if one or several sound cards are present on the host system, and the corresponding device has been selected in the Hardware IO configuration dialog.

12.2 Hardware devices

12.2.1 Input & output devices

The FLUX:: MiRA:: allows for different input and output devices. The following sections describe the available options.

This setting lets you choose amongst a selection of devices, depending on your particular hardware configuration.

None

This turns off hardware input and output altogether. This is the recommended choice if you do not want to take advantage of MiRA's built-in audio capabilities, for example if you're working with a SampleGrabber inside a DAW or Avid Venue console setup. With some sound cards that aren't multi-client capable - meaning only one program can access it at once - disabling I/O is necessary to continue using another program simultaneously.

Your soundcard

Any installed soundcard(s) will be listed here. Under Windows, it might appear several times, in which case be sure to select the native ASIO driver for performance, not an emulated driver which be labeled something like ASIO DirectX Full Duplex Driver, Generic Low Latency ASIO Driver or similar.

12.2.2 External sampling rate

Allows the Flux:: MiRA:: to follow the sample rate settings of the attached audio interface.

12.2.3 Sampling rate

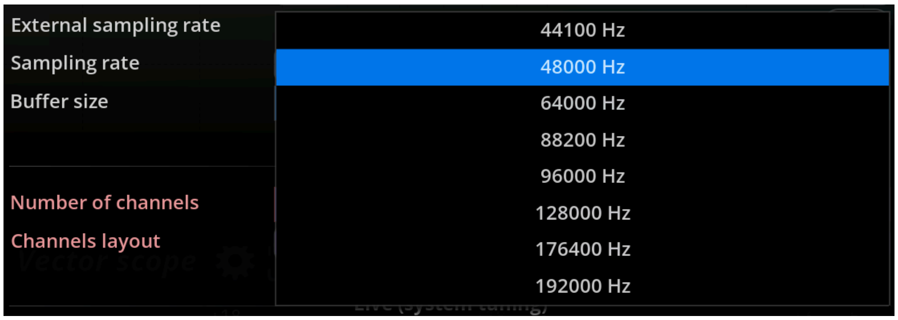
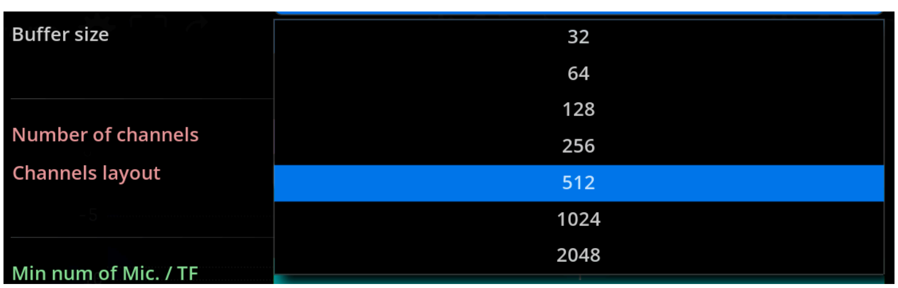


Figure 12.2: Available sampling rates (hardware specific)

Sets the sampling rate used internally by the application. When a hardware device is selected, be sure to match this to the sampling rate set in the application panel of your soundcard control panel. We deliberately chose not to employ resampling, which, in our opinion, has no place in a measurement instrument. Instead, we generally advise you to set your soundcard's sampling rate to 44.1k or 48k, which covers the entire audio hearing range (20-20kHz). Increasing the sampling rate above these values increases the processing power required to carry out the computations without any benefit for most practical applications.

12.2.4 Buffer size

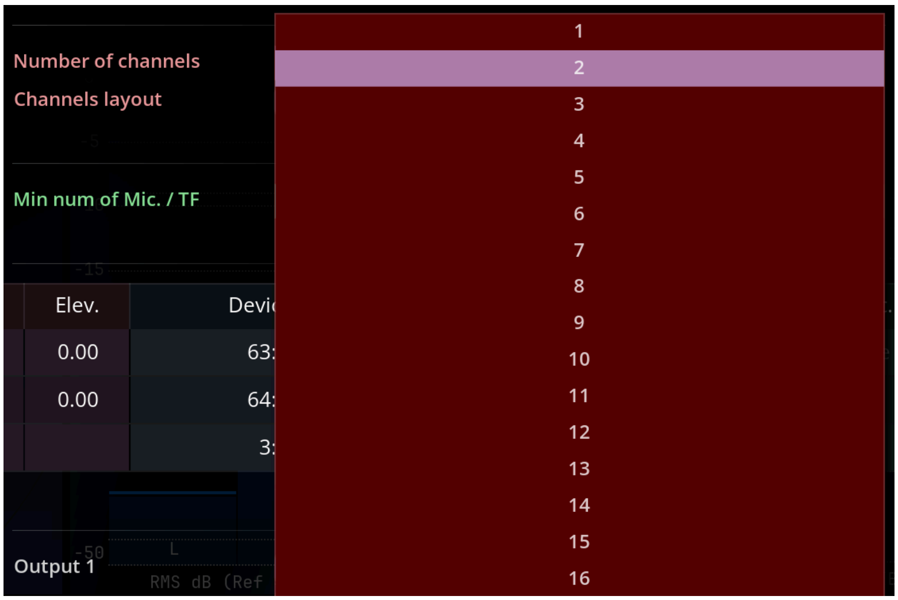


Displays the current soundcard I/O buffer size. Depending on your soundcard, you might be able to change this to a different value directly in FLUX:: MiRA:: without opening its control panel beforehand. Smaller buffer sizes lead to a shorter latency between incoming audio, display updates, and audio output. This setting is certainly not as crucial as in the context of live sound processing, so there is no need to go down to extremely small values here, as this only increases the system load without offering any practical advantage.

Keep in mind a display refresh rate of 60Hz means one frame lasts for approx. 16ms, which is a bit longer than one 512 buffer at 44.1kHz, so the display will always lag less than one frame after the audio with such a setting.

12.3 Input (Reference)

12.3.1 Number of channels



Selects the maximum number of channels to be used by the application, or equivalently the number of channels in the application I/O bus. You should set this according to the source material format you want to analyze and visualize. This determines notably how many real-time curves are displayed in the Spectrum analyzer 15.1 view, whether the Surround scope 19.1 is displayed, etc.

> **!** Important
>
> The FLUX:: MiRA:: supports up to 24 channels of audio.

12.3.2 Channel layout

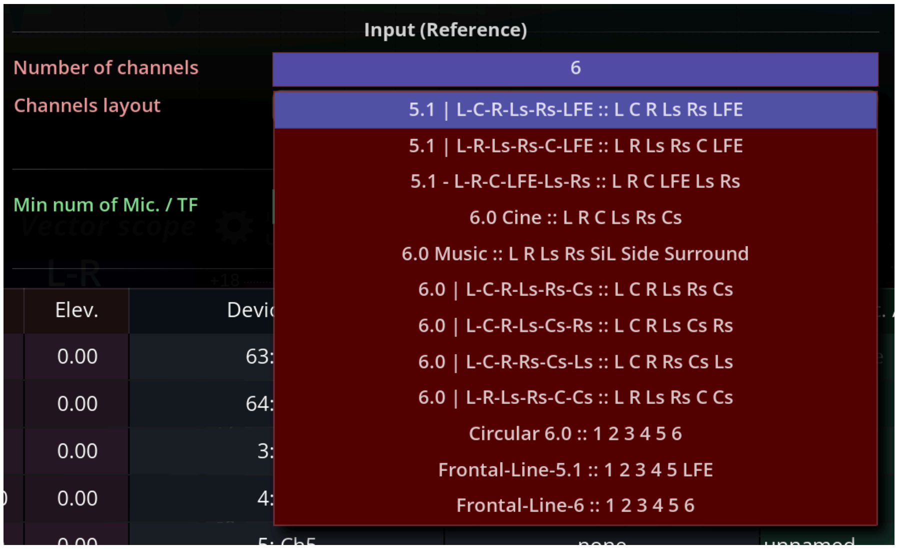


Figure 12.3: Reference configurations available with 8 max. channels

Choose the channel layout for the reference input stream. New channel layouts can be imported by using the menu `File>Import IO setup`. Note that only `.flux_io` files are supported at the moment. All imported IO configurations are stored in the FLUX:: IO config folder, located at `/Users/user_name/Library/Application Support/FLUX/IOConfigs` on macOS and at `C:\User\/...\AppData\Local\FLUX\IOConfigs` on Windows.

12.4 Live (system tuning)

12.4.1 Max number of mic. / transfer function

Defines how many input channels are fed to the capture engine.

12.5 Channel specification

The channel specification table allows to specify options per channel:

- The name of the channel - display only
- The position (Azimuth, Elevation) - display only
- The routing (regarding the input and output device) - display only

- The name in the transfer function window
- Define whether the channel is used as a reference or a microphone channel in the transfer function window.
- Phase inversion

> **! Important**
>
> This table is shared between the real-time mode and the transfer function mode of the application. Information relative to the input reference is independent from the information relative to the transfer function measurement tools and the other way around.

12.6 Signal generator

12.6.1 Output

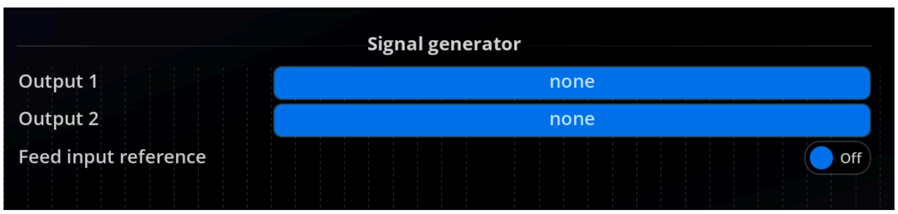


Figure 12.4: Example of an output channel routing (hardware specific)

Selects one or two physical channels to which the 31 output should be sent.

> **Tip**
>
> In the case of stereo output, the signal is identical on both channels. This is provided as a facility for soundcards with minimal routing capabilities, and to avoid using a Y patch cable.

12.6.2 Feed input reference

Best practices in terms of function transfer measurement advice include making a physical loop between an output and an input of the audio interface to feed the noise generator into the reference input of the MiRA::.

In many scenarios, it is much easier to create a “software” loop. Activating the feed input reference sends the output of the noise generator directly to the reference input without the need for a physical loop.

Warning

Beware that, while being a handy option, we recommend using a physical loop as often as possible. The main drawback of the "software" loopback is that the latency of the sound card skews the delay measurements.

13 Top Bar Menu

The Top Bar Menu provides access to additional functionalities beyond the main view of the app. It includes the main MiRA drop-down for accessing IO, UI, and main settings; the file drop-down for storing, opening, and creating workspaces; the edit drop-down for editing workspaces and layouts; and the view drop-down for changing the current visible layout.

13.0.1 Main Menu

Action Name	Comment
About MiRA	Display the credits of the application
Main settings	Open the main settings of the app
IO settings	Open the IO settings
UI settings	Open the UI settings

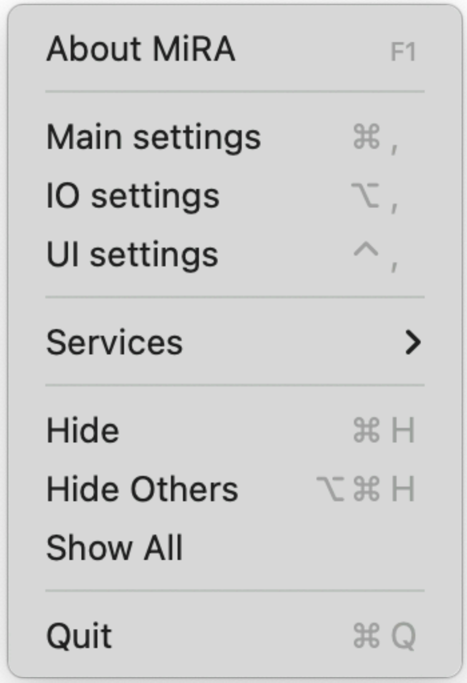


Figure 13.1: Main app menu

13.0.2 File Menu

Action Name	Comment
New workspace	Create a new empty workspace file at the place specified by the user
New workspace from factory template	Create a duplicate of the default workspace into a file at the place specified by the user
Import factory template layouts	Import the default layout into the current workspace
Open workspace	Open a workspace file
Save workspace	Save the current workspace
Save workspace as	Save the current workspace into a new file
Reload workspace	Reload the current workspace and erase all unsaved modifications

Action Name	Comment
Import IO Config	Allows to import `.flux_io` files. Multiple files and folders can be imported simultaneously.

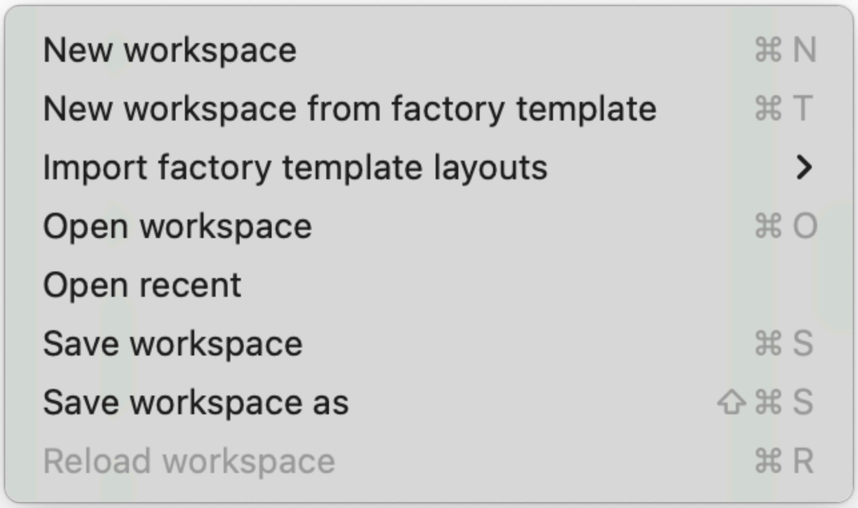


Figure 13.2: File menu

13.0.3 Edit Menu

Action Name	Comment
Show workspace toolbar	Activate the edition mode
Layout	Sub-menu relative to layout edition
Refresh network connection	Rescan network for sample push instances
Toggle generator on/off	Activate or deactivate the signal generator
Reset all meters	Reset all meters, including peak hold and integration values.
Enable advanced editor	Switch the layout editor mode to the advanced mode

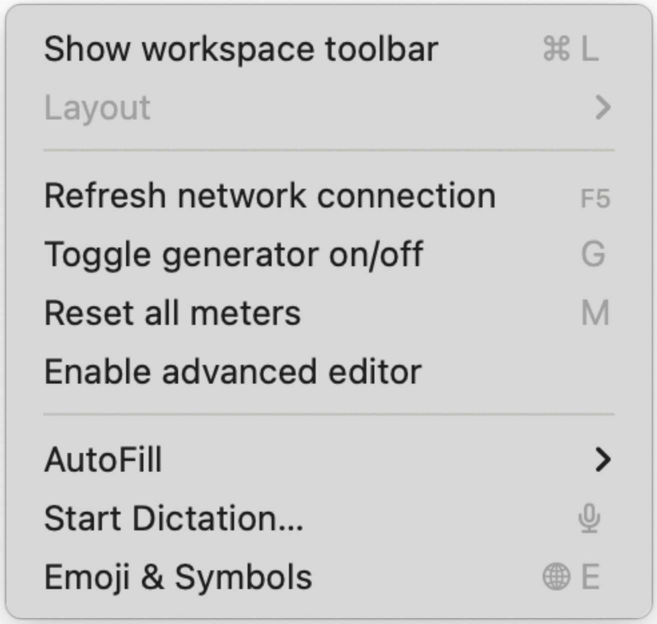


Figure 13.3: Edit menu

Layout Menu

Action Name	Comment
New layout	Create a new layout
Edit current	Edit the currently displayed layout
Rename	Rename the current layout
Duplicate	Duplicate the current layout
Delete	Delete the current layout
Add scope	Open a menu to add a new scope to the current layout
Add...	Select a scope to add to the current layout
Remove all	Remove all scopes from the current layout

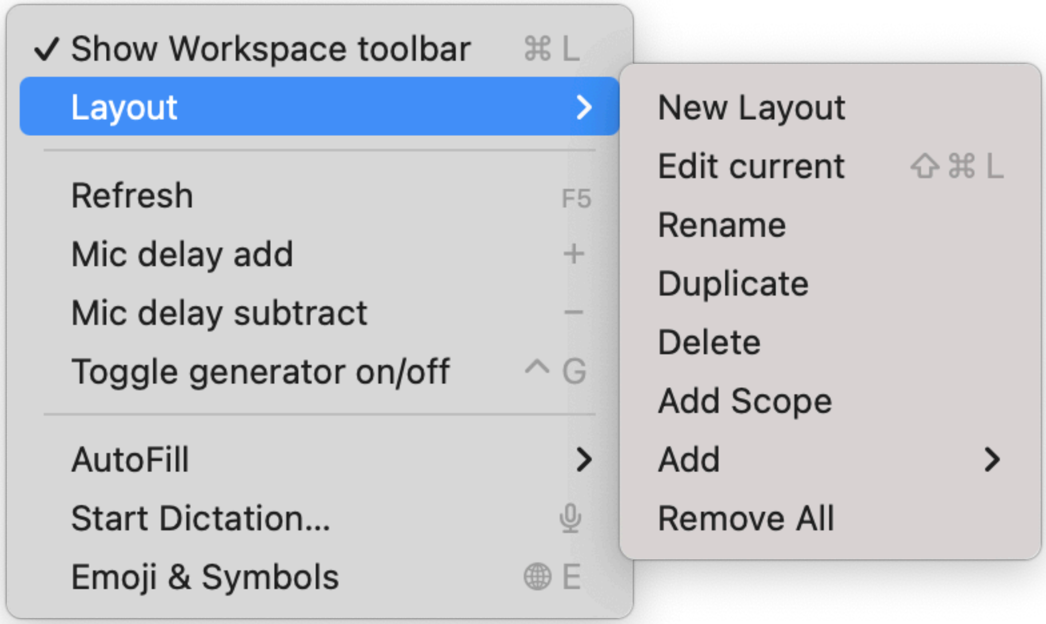


Figure 13.4: Layout menu

13.0.4 View Menu

Action Name	Comment
Layout	List all the layouts of the current workspace. Clicking on one layout switch to it.
Close setup	Close the currently opened setup window
Splash MiRA	Display the splash screen

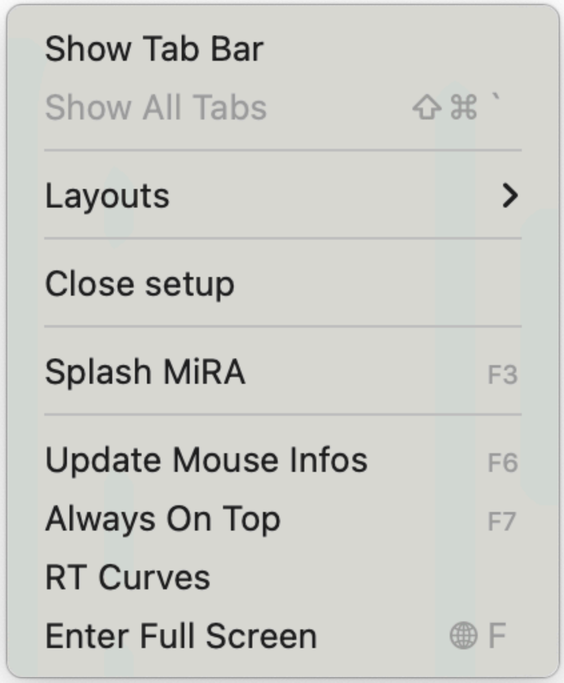


Figure 13.5: View Menu

13.0.5 Capture Menu

Action Name	Comment
New session	Create a new session
Import session	Import a session
Export session	Export selected session
Consolidate capture calibration in session	Consolidate the current session by copying used calibration curves inside.
Duplicate session	Duplicate selected session
Delete session	Delete selected session
Capture in new session	Create a new capture in a new session
Capture with pink noise in new session	Create a new automated capture in a new session using a pink noise
Capture with sweep in new session	Create a new automated capture in a new session using a sweep

Action Name	Comment
New capture	Create a new capture from current measurement
New automated capture with pink noise	Create a new automated capture using a pink noise
New automated capture with sweep	Create a new automated capture using a sweep
Capture computed curve	Crete a new capture from the computed curve
Invert capture	Invert the magnitude response of the capture
Import capture	Import a capture in the selected session
Export captured audio to .wav file	Export the selected capture to audio
Delete capture	Delete the selected capture
Update capture setting	Update the capture settings. It is very useful to change the spectrum type or the analysis block size after the audio capture process.
Increase the delay by one sample	Add one sample of delay to the selected capture
Decrease the delay by one sample	Substract one sample of delay to the selected capture
Increase the delay by ten samples	Add ten samples of delay to the selected capture
Decrease the delay by ten samples	Substract ten samples of delay to the selected capture
Find delay	Compute the delay between the recorded reference and selected capture
Regenerate color	Regenerate selected capture color
Edit notes	Edit notes of the selected capture

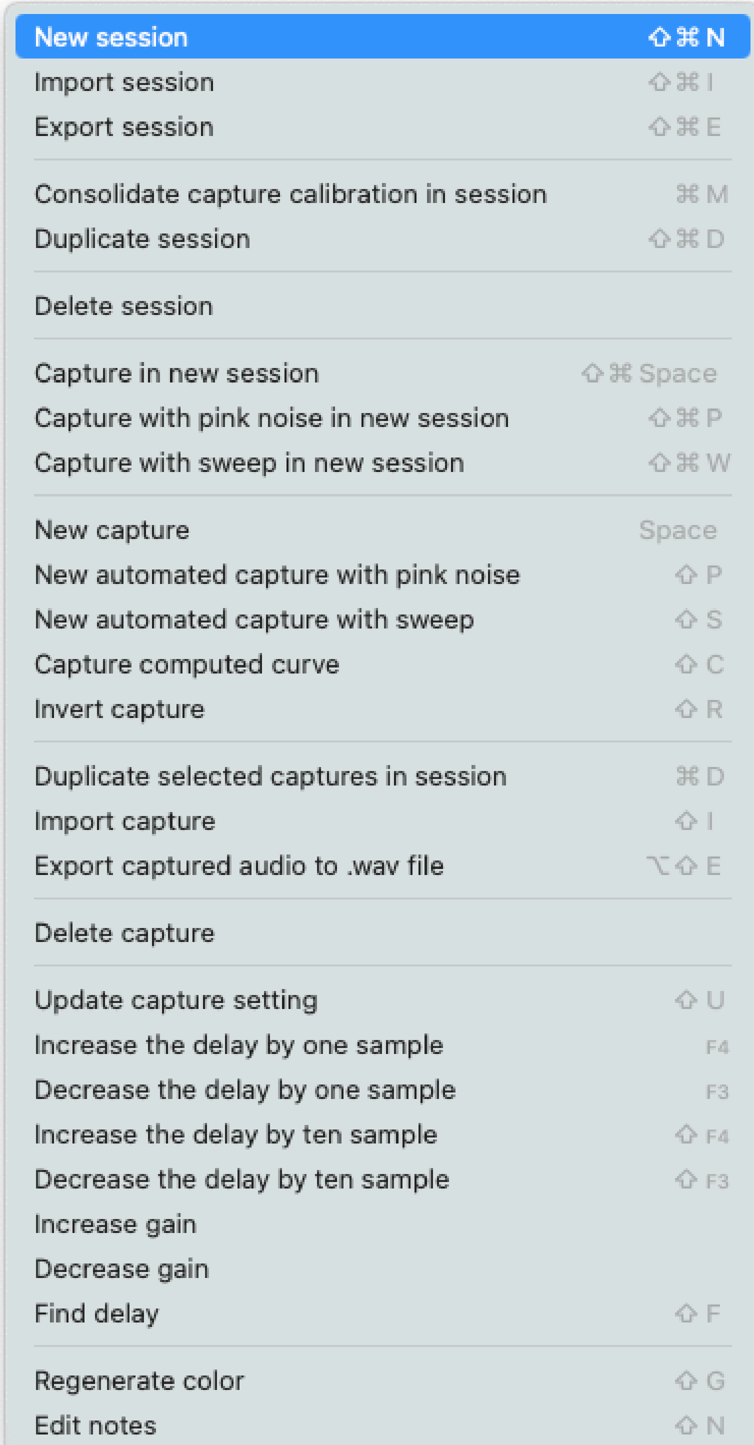


Figure 13.6: Capture menu

14 SamplePush Technology - SampleGrabber

14.1 Principle of operation

The FLUX:: MiRA:: app allows for complete separation of signal acquisition from analysis for maximum flexibility. To make this possible, FLUX:: has developed a proprietary solution to send audio through a local network. It is called **SamplePush** and it leverages the usage of ZeroConf/Apple Bonjour protocol to make the whole setup a breath. Finally, the FLUX:: MiRA:: standalone application receives the sample feed(s) and analyzes them.

At this point in time, there are two products that have the capability to send audio samples using SamplePush : the **SampleGrabber** plugin and **SPAT Revolution**. The rest of this page will focus on the SampleGrabber only. Please consult the documentation of SPAT Revolution for more details about the possible integration with MiRA.

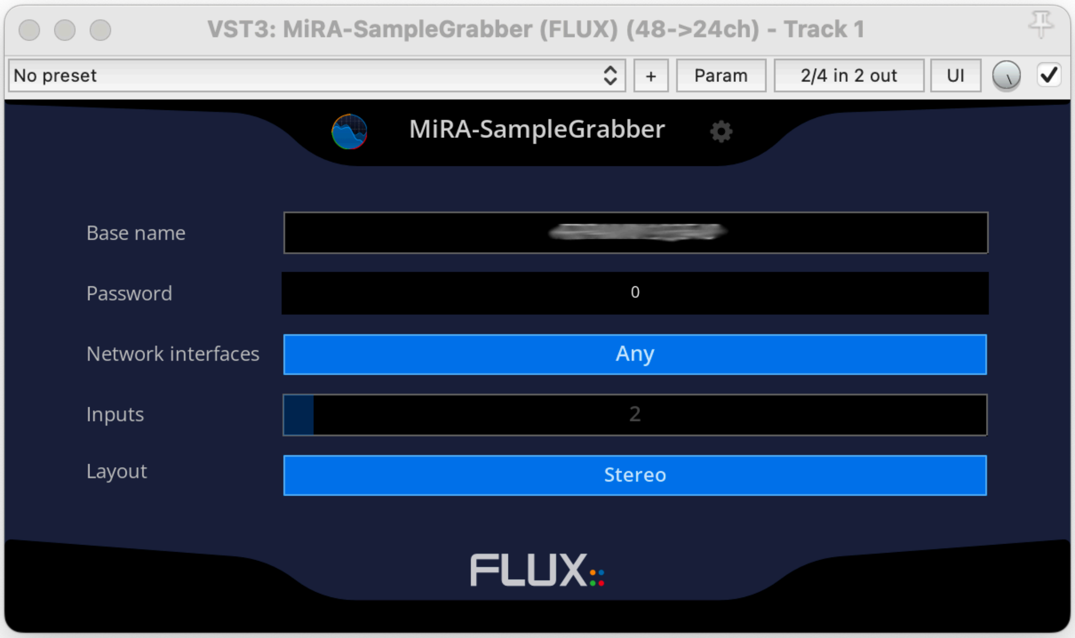


Figure 14.1: UI of SampleGrabber

SampleGrabber is a multichannel capable plugin, available in all common formats (VST, AU, RTAS and TDM), whose channel configuration is set automatically, or by clicking the icon and setting the desired channel count in the I/O submenu.

The FLUX:: MiRA:: application displays in the Audio source menu a list of SampleGrabber instances found on the network. Each instance is identified by the network name of associated computer on which it is running on. Clicking a name on the list will select the corresponding SampleGrabber for input.

> **Note**
>
> You can insert up to 64 instances of SampleGrabber plugins inside one same DAW, and up to 64 FLUX:: MiRA:: instances can be connected to any SampleGrabber instance over the network. A SampleGrabber can be connected to up to 64 FLUX:: MiRA:: instances over the network. *We do, however, recommend limiting the number of instances in order to avoid saturating the network.*

14.2 Network configuration

Network configuration is completely automatic and transparent, thanks to the use of the ZeroConf/Apple Bonjour protocol. Should you encounter any problems with your connection, we advise you to check whether the UDP port range from 46000 to 46064 is opened in your firewall, for both incoming and outgoing connections.

Audio transport requires approximately 1.4 Mbps for each channel at a sample rate of 44.1kHz, whereas a 5.1 configuration at 96kHz demands a little less than 20 Mbps. A properly functioning standard Ethernet 100 Mbps network should, therefore, be more than sufficient to handle most scenarios.

> **Note**
>
> The above bandwidth requirements naturally do not apply when using both SampleGrabber and FLUX:: MiRA:: on the same machine.
> Please check with your network administrator if you have any bandwidth issues and/or special requirements.

14.3 Password

An optional password, which is a simple 4-digit number, allows you to apply light encryption to the audio stream for secure transmission over the network. It is set to 0 by default, which turn off encryption; in this case, no additional action in the FLUX:: MiRA:: application is required on your part.

If you wish to employ and define a password in SampleGrabber, please enter a matching value in the SampleGrabber menu of the FLUX:: MiRA:: application in order to be able to decrypt the incoming stream.

Please note that the security level provided by this encryption is mild, and is only intended to protect from anyone eavesdropping your audio stream inside the internal network. It is not intended as a substitute for conventional network security practices and measures such as software and hardware firewalling, etc.

Part VI

Audio Analysis Scopes

Scopes are a small part of the graphical user interface that is used to build the layouts. Each scope has a very specific purpose. Most of the time, it is an audio analysis tool, but it can also simply be a helper interface with some quick access buttons to improve the workflow.

List of scopes

The scopes' list is accessible here.

Scope header

All scopes have a header that contains some buttons. Only the common buttons will be addressed here; otherwise, refer to the documentation section dealing with the specific scope.

- The small cog opens the settings panel of the scope.
- The four corners icon allows the display of the scope in full screen.
- The play button toggles the real-time display on and off.

> **Note**
>
> The real-time display is a global control. If you change its state on one scope, it will affect all the other ones.

Scope presets

Inside the **settings panel** of a scope, you will find the scope's presets. Each scope can have many different presets that can be stored and recalled. A preset stores all the settings of a scope.

While the scope settings are stored in the layouts and in the workspaces, presets are useful for sharing scope settings between workspaces.

Scope appearance

Figure 14.2: Background colors

Inside the settings of each scope, you can find a "background" category. A scope can either follow one of the globally defined colors (see UI menu) or use a custom solid/gradient color.

The **Background type** option defines whether a solid or a gradient background should be used for the scope. Use "global"-prefixed options to refer to the global app theme. Use "custom"-prefixed options to override the global options for this scope.

15 Spectrum Analyzer

15.1 Presentation

A spectrum analyzer's global principle and purpose is to transform an incoming signal, which is basically a series of amplitudes taken at successive points in time, into a series of values versus frequency. Transforming an audio signal onto a frequency scale is indeed of great interest in a wide range of tasks, especially because it provides a global, perceptually meaningful and precise picture of the audio contents.

The display represents the so-called magnitude spectrum of the incoming signal, which is a two-dimensional curve of the magnitudes of the signal taken at frequencies ranging from 0 (DC) to half that of the current sampling rate (or Nyquist frequency in signal processing jargon). This is probably the most commonplace and most easily understood spectrum analyzer visualization, and the place where you should start most of the time when you want to inspect the frequency content of your audio material.

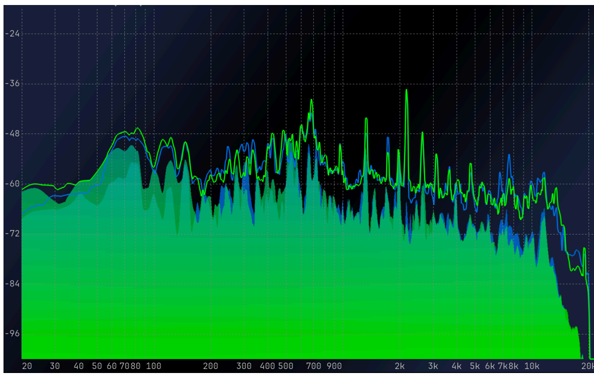
15.2 Channel Selection and Summing

Above the spectrum analyzer, there are special buttons for channel selection:

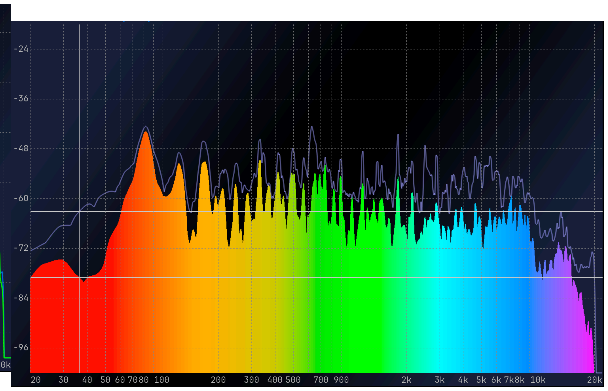
- Numbers associated with input channels
- A "plus" symbol under each channel number

When an input channel is **on**, its spectrum is displayed in the spectrum analyzer. When the "plus" button of a channel is **on**, it feeds its signal to the summation curve. Several channels can be sent to the summation curve.

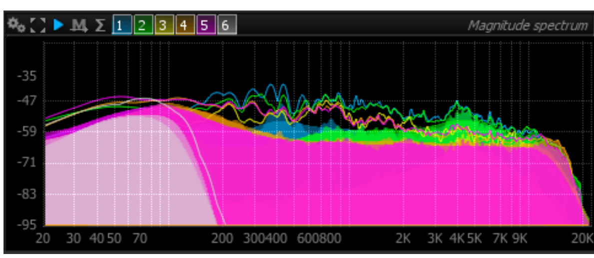
> **!** Important
>
> This means that you can display both the sum or individual channels in the same spectrum analyzer. To remove the sum plot, uncheck all the "plus" signs under the channel



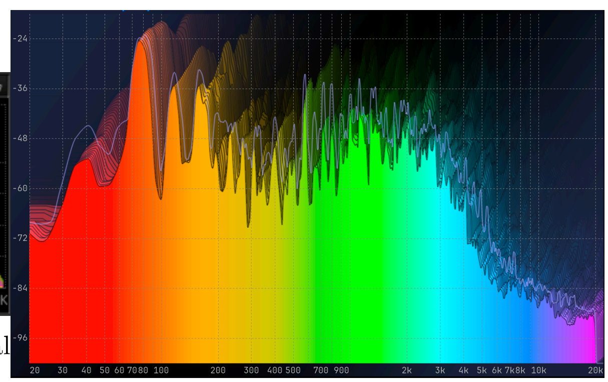
(a) Magnitude spectrum of a stereo signal with summing disabled, max and smoothed curves enabled



(a) Magnitude spectrum of a 5.1 surround signal sum with max and smoothed curves enabled



(a) Magnitude spectrum of a 5.1 surround signal with summing disabled



(a) Magnitude spectrum with “Slide” option enabled (Real time waterfall) numbers.

15.3 Settings

15.3.1 IO

Name	Description
Use input (reference) layout	Define if the number of channels displayed by the meter reflects the current input reference layout or the number of channels of the system tuning inputs.

15.3.2 Range

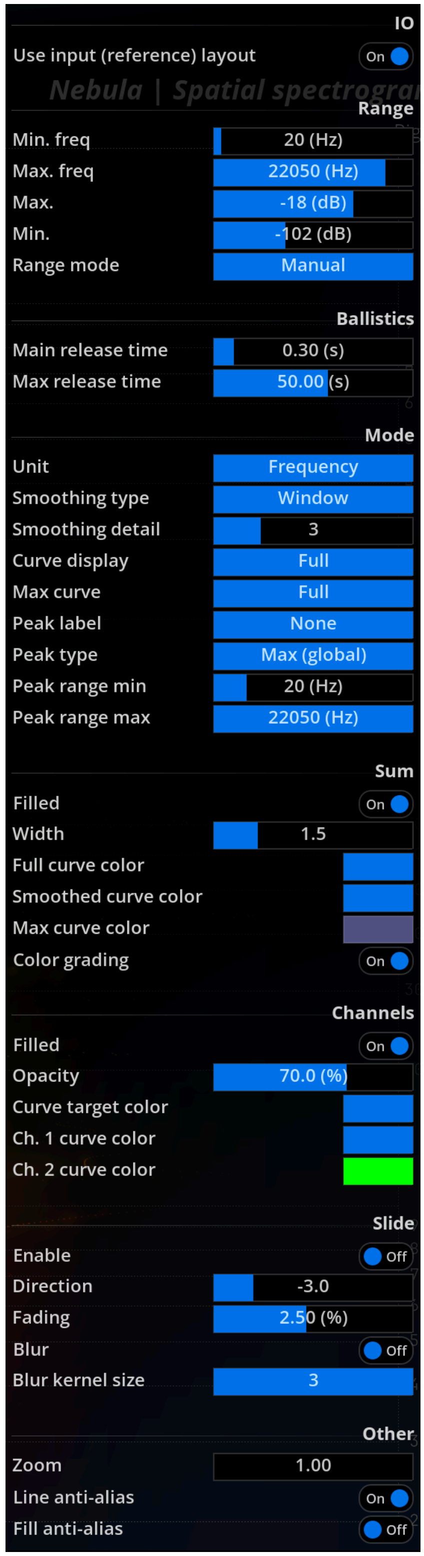
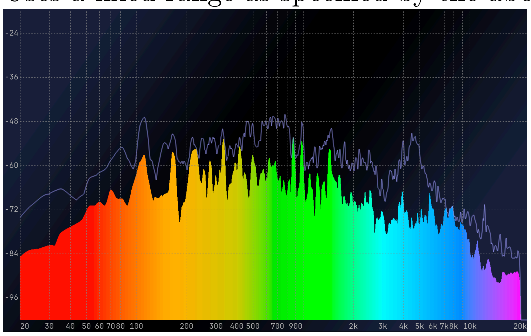
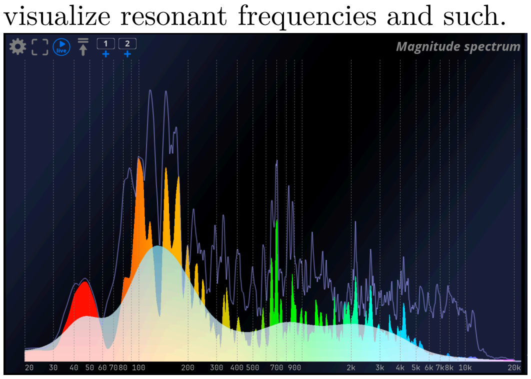


Figure 15.5: Magnitude spectrum setup dialog

Display range can be switched from a fixed reference interval to one that automatically adjusts to the current range of spectrum magnitude values. The latter is useful as a set and forget setting and works well to display the most vertical detail, at the expense of losing the ability to visually compare the current values to a reference level.

Name	Description
dB Min / dB Max	Sets the minimum and maximum magnitude to display, in decibels. This is visible the range of the display that is taken into account when auto-range is off. Default range is -18dB (min) to -114dB (max).
Range mode	Default is *Manual*.
Manual	Uses a fixed range as specified by the above settings. 
Auto	When engaged, auto-range continuously adjusts the display to the current range of the data. [1]
Compressed	The range is defined by dB Min/Max values, and the Y-axis is also compressed in lower range. This can bring out peaks and valleys in the spectrum to better visualize resonant frequencies and such. 
Compressed / Auto | Combines *Compressed* and *Auto* modes.	

[1] A slight envelope is applied to the auto-range values in order to improve legibility, avoiding the display to follow every minor change. Peaks are always registered however, as these provide valuable information that should not be missed.

15.3.3 Ballistics

The ballistics settings control the curve display update speed.

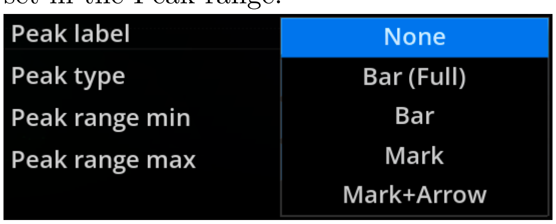
Name	Description
Release time	The release time determines how fast the main curve falls back to zero. Default is 300ms.
Max release time	The controls the release time of the optional *Max* curve, which serves to display the medium-to-long term tendency of the magnitude spectrum. Longer times mean curve maxima/peaks will be seen for a longer period. Default is 50 seconds.[2]

15.3.4 Mode

Name	Description
Smoothing type	Switches between Window (default) and various per-octave smoothing types. When *Window* type is selected, a sliding window average of adjustable width is applied to the curve, which results in more or less frequency detail being removed, depending on the Smoothing detail setting. When any of the Octave types are selected, the average of the spectrum over the corresponding ISO bands is displayed, as a series of horizontal bars. The following series are available: - Octave - 2/3 octave - 1/2 octave - 1/3 octave - 1/6 octave - 1/12 octave
Smoothing detail	Controls the amount of frequency detail of the smoothed curve, when using window smoothing. The value roughly corresponds to the maximum number of valleys and peaks that can stand out the smoothed curve. A low value lets the global tendency of the amplitude spectrum pass through, while values above 20 or so preserve more detail such as harmonics and sharp equalizer cuts and boosts. Default is 3.[3]

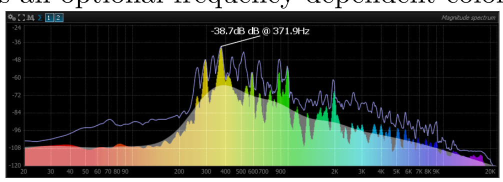
[2] The attack time is zero so the curve display reacts instantaneously to a rising amplitude.

[3] This curve acts as a kind of zoom-out control, as it shows the global frequency content of the signal, leaving out details such as harmonic peaks and variations imputable to transient and noise components. Typical uses for this curve are to monitor the global frequency balance of a mix and to visualize the influence of broad equalizer corrections on the mix.

Name	Description
Curve display	 Toggles between the following curve display modes: - **Full**: main curve only (no smoothing). - **Smoothed**: smoothed curve only. - **All**: both unsmoothed and smoothed curves. [4]
Max curve	 The max curve employs a much longer release time compared to the main curve, and, as such, registers short peaks much more easily. The max curve setting controls its visibility and whether smoothing is applied: - **None**: curve not displayed. - **Full**: visible, unsmoothed. - **Smoothed**: visible, smoothed. [5]
Peak type	 This setting controls the manner in which spectrum magnitude peaks are computed: - **Max (global)**: compute a global maximum over the entire spectrum range. - **Max (user)**: compute the maximum across a user defined portion of the spectrum set in the Peak range.
Peak label	 Determines the appearance of the peak display: - **None**: peaks are not shown. - **Bar (Full)**: vertical bar at current peak located at current frequency. - **Bar**: vertical bar from base to peak value. - **Mark**: text box indicating peak value, in dB, and frequency (Hz) at peak location. - **Mark + Arrow**: same as above, with text at the top of the display and arrow pointing at peak location. This is the most precise indication, but it takes up more space.
Peak range	Used in combination with the *Max (user)* Peak type setting, this defines the minimum and maximum frequencies to take into account when computing the peak.

15.3.5 Summation

These settings allow you to modify the appearance of the curves in channel sum mode.

Name	Description
Filled	Toggles whether the main curve is drawn as a solid-color fill or a plain line. Default is on.
Width	Thickness of the pen used to draw the curve lines, in pixels. Default is 1.0. [6]
Full curve color	Color of the pen used to draw the main, full-detail, unsmoothed curve.
Smoothed curve color	Color of the pen used to draw the smoothed curve.
Max curve color	Color of the pen used to draw the max curve.
Color grading	Applies an optional frequency-dependent coloring to the main channel-sum curve.  Magnitude spectrum with color grading enabled. [7]

15.3.6 Channels

This group of settings controls the appearance of curves when channel sum mode is disabled. There is one Ch.N curve color setting per channel, so you can fine-tune the color scheme employed if you wish to do so.

Name	Description
Filled	Controls whether channel curves are drawn as a solid color fill or a plain line.
Opacity	Controls the opacity of the fill when *Filled* is enabled. 100% gives a fully opaque fill, lowering this value makes the curve fill more transparent.
Channel curve color	This setting controls the color of the curve corresponding to the nthchannel, when summation mode is disabled.

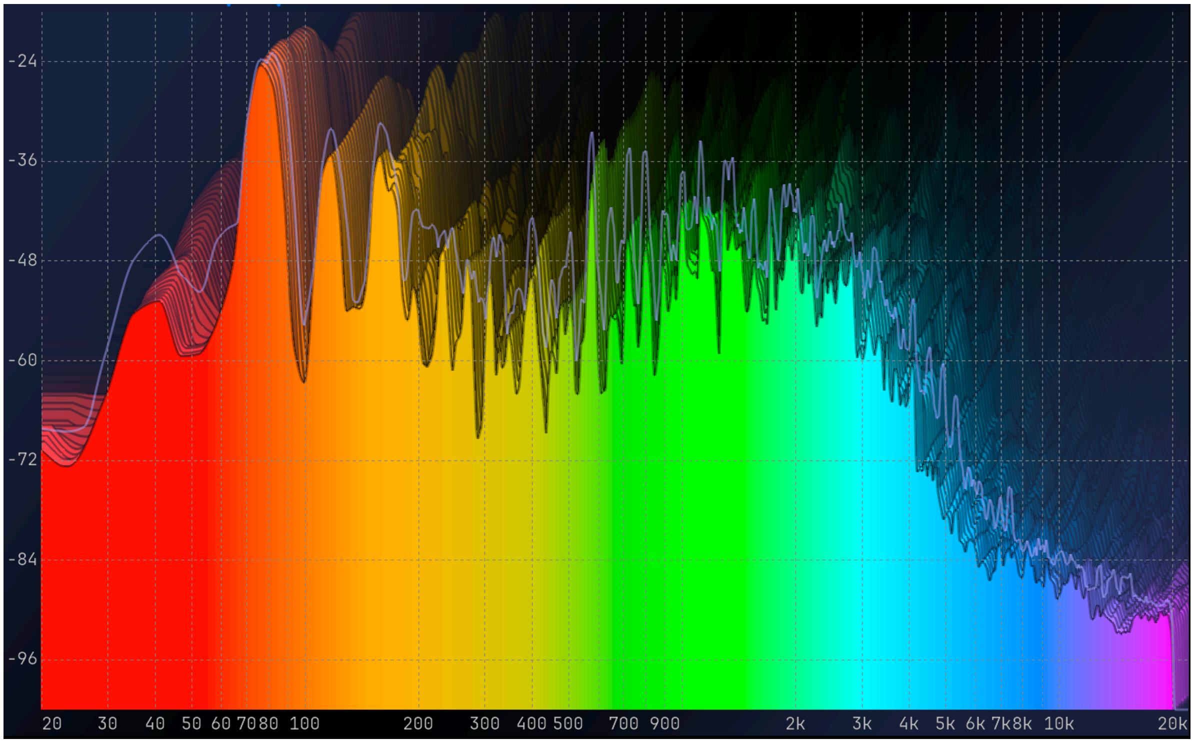
[4] Selecting one of the first two modes is recommended to avoid display clutter when comparing several channels and/or captures.

[5] The max curve is never displayed for captures, as it would be the same as the main curve, since this type of curve does not evolve in time.

[6] This setting also affects individual curves when channel sum mode is disabled.

[7] When enabled, any of the above fixed color settings are overridden.

15.3.7 Slide (Real Time waterfall)



Name	Description
Enable	Enable/disable the slide mode.
Direction	Define the sliding Direction. From -5 to 5. Default is 0.
Fading	Controls display persistence, *i.e.* the "fade to black" amount for a frame. Lowering this value retains past particles longer, whereas increasing this makes them disappear faster.
Blur	Enable / Disable sliding blur.
Blur Kernel Size	Controls the radius of the blur effect applied to past particles. Particles are "smeared" more and more as they become older, depending on this setting. Naturally, a bigger value increases the smearing, at the expense of processing power. [8]

15.3.8 Other

[8] Choosing the value for this setting is really a matter of taste, although please keep in mind values that above 5 will require a sufficiently powerful graphics card in order to maintain a responsive display.

Name	Description
Zoom	This setting allows to check and change the current X-axis zoom level. Default is 1.0, which corresponds to the whole frequency spectrum. Zooming with the mouse is the preferred way, as it offers more control.
Line anti-alias	Activate the graphical anti-aliasing of the outline of curves.
Fill anti-alias	Activate the graphical anti-aliasing of the curves.

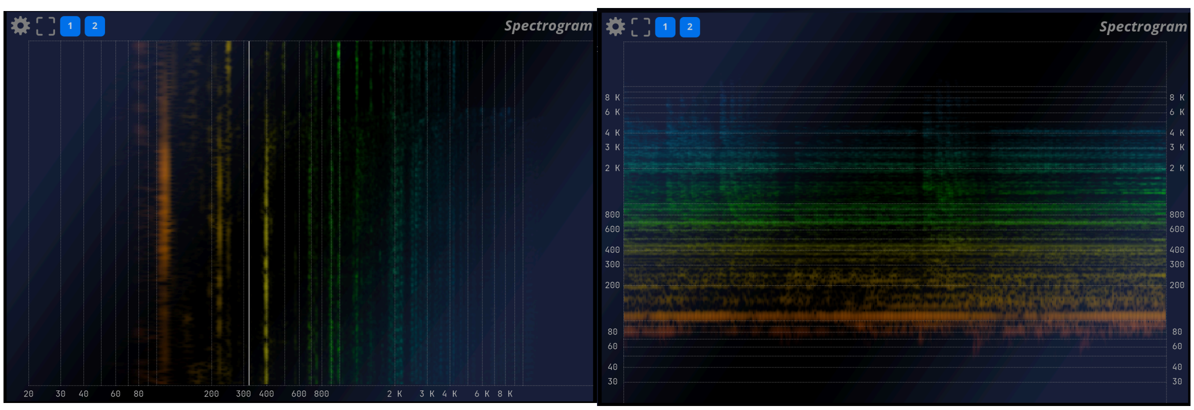
16 Spectrogram

16.1 Usage

The spectrogram is a two-dimensional view of the evolution of the signal's spectrum over time, i.e. a frequency (Y-axis) versus time (X-axis) plot (or the invert, depending on the direction setting), with the magnitude modulating the color and intensity of the pixels.

A spectrogram can be computed using the STFT (short-term Fourier transform) as well as other means. It serves as a useful tool to get a global picture of how the frequency content of a signal changes over time, and eases identification of its structure. Broadband noise appears as background, a pure tone as a horizontal line, and a transient as a vertical line.

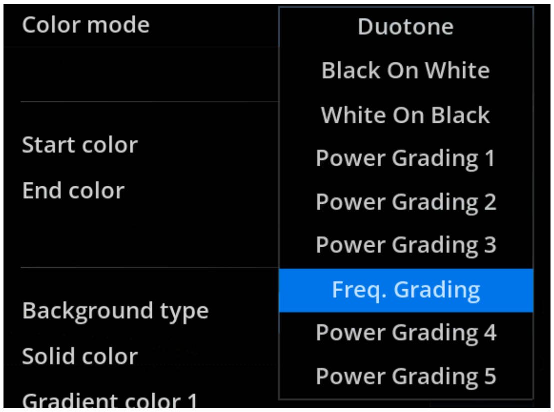
Harmonic content appears as horizontal groups of parallel lines and vertical bars respectively, etc.



16.2 Settings

16.2.1 Spectrogram setup

Name	Description
Direction	 Defines the scrolling direction of the spectrogram.
Log Gain	Toggles logarithmic scaling of the magnitude spectrum on and off. Default is on. When enabled, the magnitude at a given time-frequency point is applied a logarithmic scaling before being converted to a pixel value. This has the effect of compressing the dynamic range, and makes low energy components stand out more, but it also decreases the contrast of the display.
Range Max	Sets the maximum amplitude spectrum value to be displayed.
Range Min	Sets the minimum amplitude spectrum value to be displayed.

Name	Description
Color Mode	 **Duotone**: In this color mode, the amplitude of a time-frequency point is mapped to a pixel using a two-color palette, set using start/end colors. **Black On White**: In this color mode, the amplitude of a time-frequency point is mapped to a pixel using a Black & White color palette with White as background. **White On Black**: In this color mode, the amplitude of a time-frequency point is mapped to a pixel using a Black & White color palette with Black as background. **Power grading 1, 2, 3, 4, 5**: In this color mode, the amplitude of a time-frequency point is mapped to a pixel using a different predefined color palette. **Frequency grading**: In this color mode, the amplitude of a time-frequency point determines the intensity of the corresponding pixel, whose color varies according to frequency.

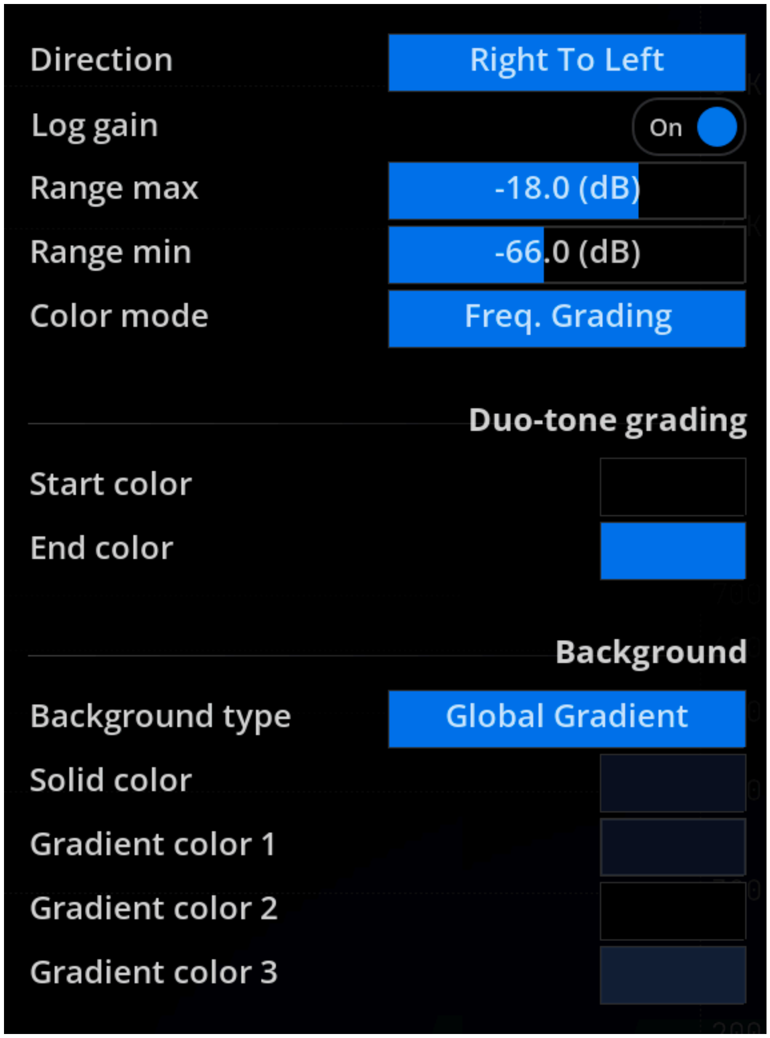


Figure 16.1: Spectrogram setup

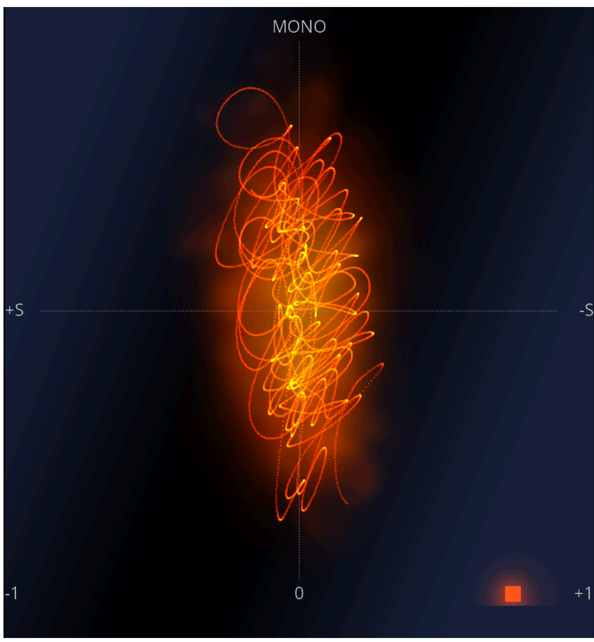
16.2.2 Duo-tone grading

Name	Description
Start/end colors	Sets the color for minimum and maximum amplitude components respectively, when color mode is set to *Duotone*.

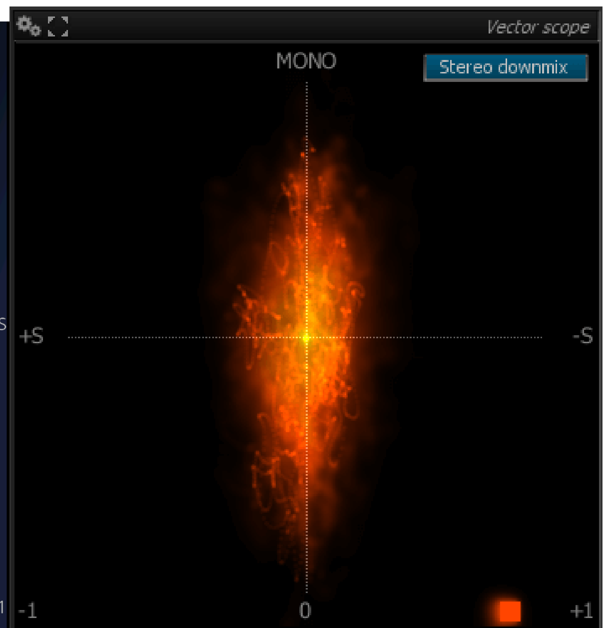
17 Vector scope

17.1 Usage

The vector scope tool is displayed when a stereo input is detected. Otherwise the display will switch to Surround scope 19.1 provided if your edition of FLUX:: MiRA:: includes this option.



(a) Vector scope display in stereo.



(a) Vector scope display in surround (with selection menu).

17.1.1 Modes in Surround :

Mode	Description
L-R	Use only Left and Right Channels.
Front	Use a stereo down mix with all front channels.

Mode	Description
Rear	Use a stereo down mix with all Rear channels.
Stereo downmix	Use a stereo down mix with all channels.
Lt/Rt downmix	Use a Lt/Rt down mix with all channels.
LR-Lfe	Use a mono summation of Left and Right + the Lfe (sub) channel.
Center-Lfe	Use Center + Lfe (sub) channel.
Front-Lfe	Use a mono summation of the front channels + the Lfe (sub) channel.

17.2 Settings

17.2.1 Mixdown

Name	Description
Mode	Select which channels or mixdown to display in the vector scope. - LR : only display the left/right channel correlation. - Front : use a summation of the speaker at the front. - Rear : use a summation of the speaker at the rear. - Stereo downmix : use a stereo reduction of the mix. - Lt/Rt downmix : use a Lt/Rt matrixing. - LR-Lfe : compare the summation of left-right channels vs. lfe. - C-lfe : compare center channel and lfe. - Front-lfe : compare front speaker's summation and lfe.

17.2.2 Display

Name	Description
Transfer Function	Change the axis to input over output instead of side over mono.
Fs	Over-sampling factor in multiples of FS, that is the incoming audio is up-sampled as necessary to reach this multiple times 48kHz. Increasing this value increases the display precision and reactivity, at the expense of a little CPU overhead.
Passes	Determine the number of drawing passes to create the particle clouds on screen. Lower values will be emphasized on individual particles, while greater values create zones. Such zones can help in reading the information provided by Nebula.
Blending	Controls the amount of particle blending with the current image, from 1 to 100%. A higher value gives more priority to the incoming audio over past frames.
Fading	Controls display persistence, i.e. the “fade to black” amount for a frame. Lowering this value retains past particles longer, whereas increasing this makes them disappear faster.
Size factor	Controls the size of individual particles with respect to screen size.
Blur kernel size	Controls the radius of the blur effect applied to past particles. Particles are “smeared” more and more as they become older, depending on this setting. Naturally, a bigger value increases the smearing, at the expense of processing power.[1]
Particle scaling	Controls the size of individual particles with respect to screen size.
Dynamic fading	Controls the display persistence by signal dynamics
Color mode	This defines how the particle color is determined: - Static color: use only particle start color (see below) - Power grading: color is modulated by overall signal RMS power - Dynamic grading: color is modulated by signal dynamics - Pw+Dyn grading: mix of the two previous modes
Particle start/end colors	Sets the particle color range to be used.

[1] Choosing the value for this setting is really a matter of taste, although please keep in mind that values above 5 will require a sufficiently powerful graphics card in order to maintain a responsive display.

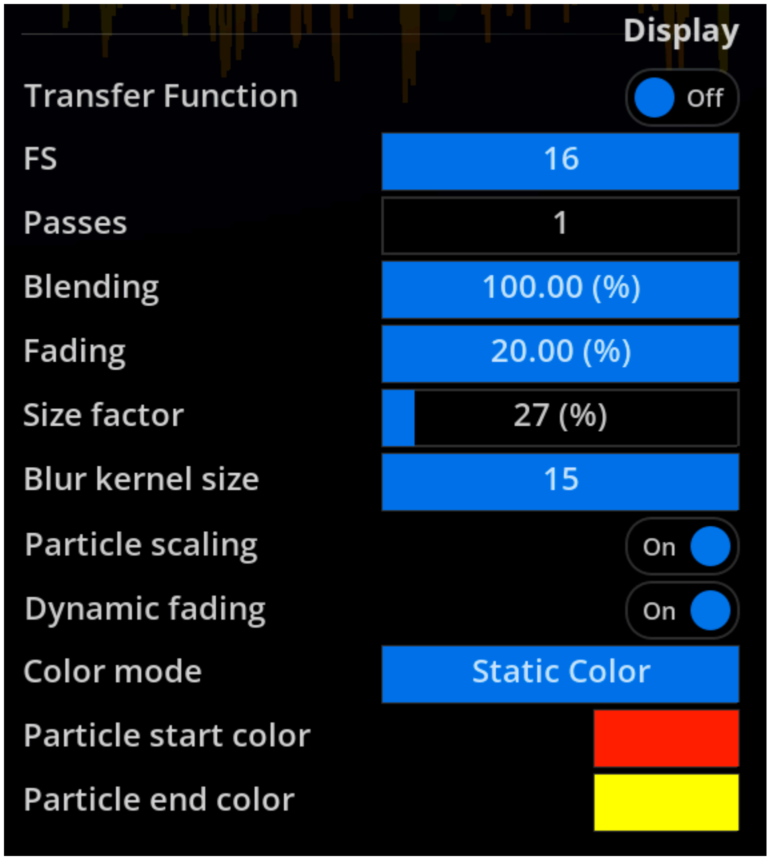


Figure 17.3: Vector scope setup options

18 Nebula (spatial spectrogram)

18.1 Principle of operation

Nebula / Spatial Spectrogram provides a unique representation of the audio material in terms of spectral content and localization in the stereo and/or surround space. It combines the functionality of a spectrum analyzer and a vector scope in a novel real-time display. As such, it is an invaluable tool to get a complete and detailed overview of your mix, which you can finely tune into many aspects to suit your particular needs and preferences. A lot of work has gone into optimizing the real-time rendering of the display, not solely for aesthetic reasons, but because we wanted the display to react instantly to all the details in the incoming audio. The idea is literally for you to be able to see what you hear and feel, and not some gross simplification wrapped into shiny eye candy, however pleasing to the eye.

The overall principles behind Nebula / Spatial Spectrogram are quite straightforward:

- At any given time, and for every frequency, the engine computes the position of a frequency in space (2D in stereo, ND for N channel surround). This position is taken as the center of gravity of the various channels, weighted by the relative amplitude of the signal in their corresponding channel.
- A projection onto an LR-spectrum plane is computed, giving a spectrum-space frame constrained to the stereo field.
- Incoming spectrum-space frames are added back to the previous frames.
- Past frames are progressively "forgotten", using blur and dimming, in order to make place for new information, and increase legibility.

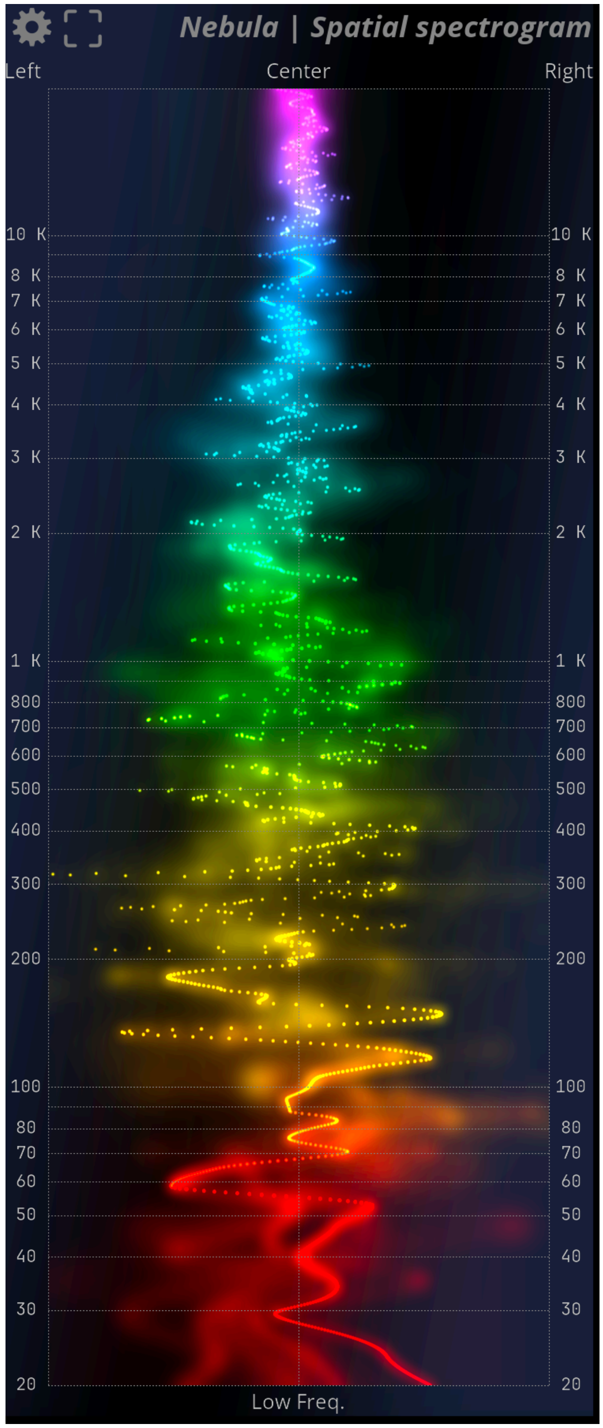


Figure 18.1: Nebula / Spatial Spectrogram display with stereo input

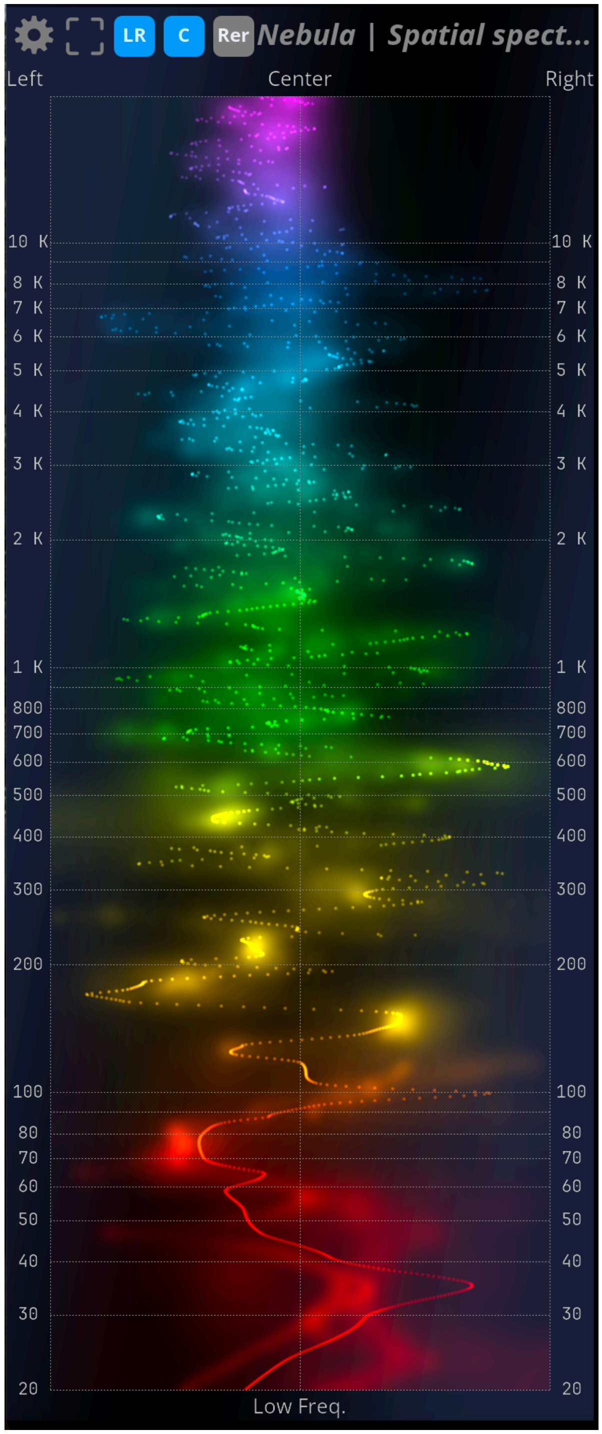


Figure 18.2: Nebula / Spatial Spectrogram display with surround input

18.2 Settings

18.2.1 Range

Figure 18.3: Nebula / Spatial Spectrogram setup options

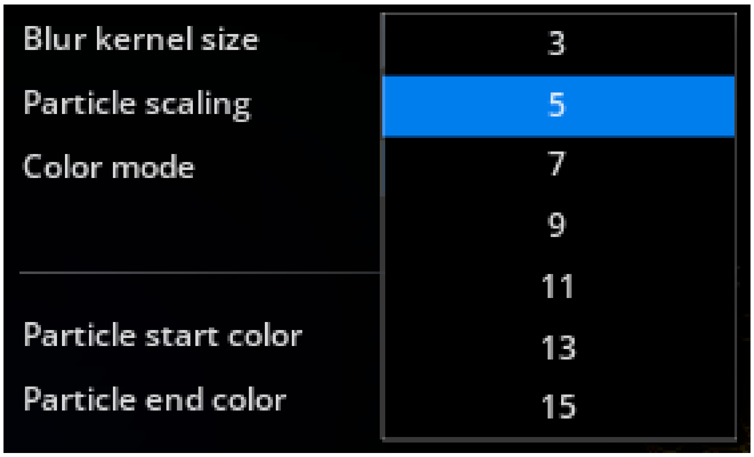
Name	Description
Min. Freq	Change the minimum frequency displayed by the scope. Default is 20 Hz.
Max. Freq	Change the maximum frequency displayed by the scope. Default is 22050 Hz.

18.2.2 Scale

Name	Description
Focus	Controls the stereo image width X-axis display range, in dB. A value comprised between ±18 and ±24dB correlates well with our abilities to perceive the stereo image. Default is ±18dB.[1]
AutoScale	This parameter controls whether the overall audio level variations modulate the intensity of the particles. In essence, when enabled, the color nuances will vary according to the relative amplitude of a frequency, allowing to monitor the relative amplitude spectrum variations. When disabled, the color will reflect the absolute audio level. You can also think of this as a kind of auto-gain setting.
AutoScale release	This controls whether color variations should be smoothed in time or not. When engaged, color variations are slowed down a bit, which makes overall level transitions more obvious.[2]
Linear blend range	Adds a constant blend amount to the particle. This ensures some particles are always blended into the image even if their original magnitude is low. A low value for this setting stabilizes the appearance of particles. With large values, more of the spectrum dynamics are taken into account, and only peaks mostly come through.
Log blending	Toggles between linear and logarithmic blending of the current particle with old particles. The default is off, i.e. linear blending, which tends to favor the display of peaks. Logarithmic blending, on the other hand, preserves more of the full dynamic range of the data, and also gives some visibility to lower levels.

18.2.3 Display

[1] Pixels outside the focus range are clamped to the view boundaries.
[2] You should enable this setting when you want to visualize quick level variations, such as those that frequently occur in movie soundtracks.

Name	Description
Passes	Determine the number of drawing passes to create the particle clouds on screen. Lower values will be emphasized on individual particles, while greater values create zones. Such zones can help in reading the information provided by Nebula.
Fading	Controls display persistence, i.e. the “fade to black” amount for a frame. Lowering this value retains past particles longer, whereas increasing this makes them disappear faster.
Size factor	Controls the size of individual particles with respect to screen size.
Blur kernel size	 Controls the radius of the blur effect applied to past particles. Particles are “smeared” more and more as they become older, depending on this setting. Naturally, a bigger value increases the smearing, at the expense of processing power.[3]
Particle scaling	Toggle-automatic adjustment of particle size with screen size. When enabled, the overall aspect of the display will remain similar even if the view size changes.
Color mode	Provides the following particle-coloring modes: - Power: the color varies according to the power of the signal in the frequency region - Dynamics: same as previous except this mode works on signal dynamics - Power / dynamics: a mix of the above - Frequency: the color varies according to frequency only, using a rainbow palette.

18.2.4 Power color grading

[3] Choosing the value for this setting is really a matter of taste. However, please keep in mind that values above 5 will require a sufficiently powerful graphics card in order to maintain a responsive display.

Name	Description
Particle start color	Sets the color to use for maximum amplitude when color mode is set to Duotone.
Particle end color	Sets the color to use for minimum amplitude when color mode is set to Duotone.

19 Nebula surround

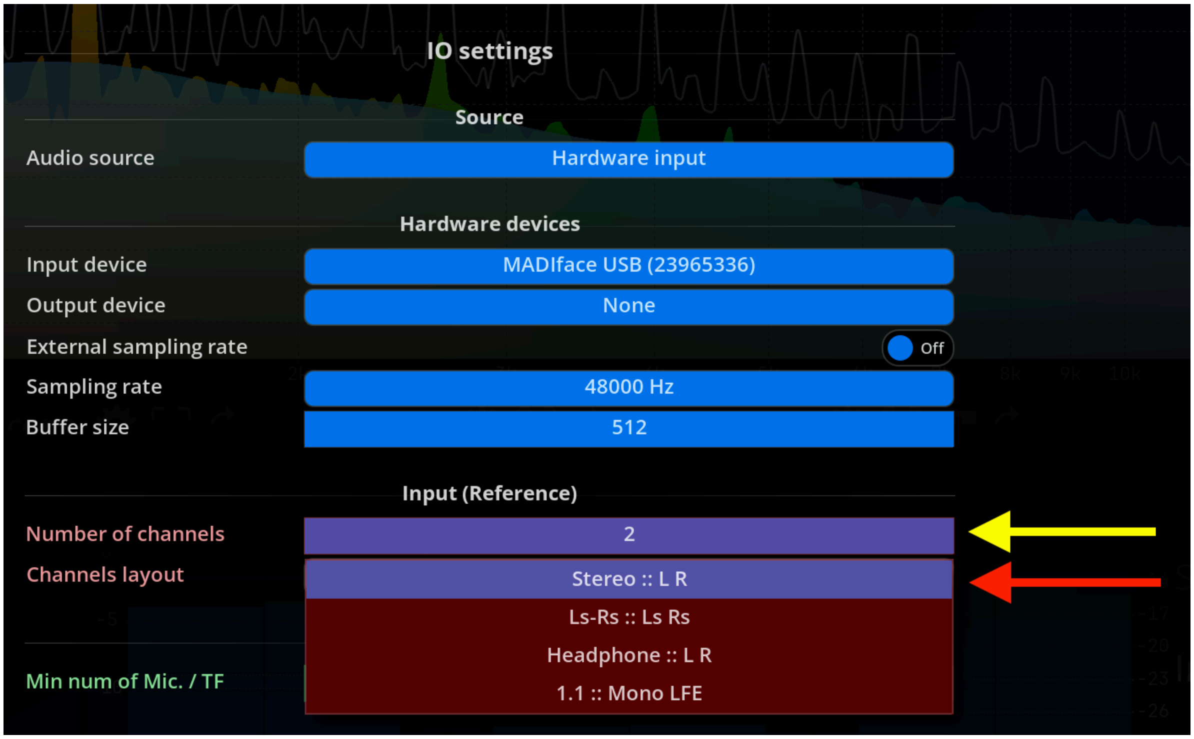
19.1 Usage

The Nebula | Surround scope displays a representation of how a surround signal's various components are distributed in a surround environment. The inner region displays the location of the signal frequency components in the selected surround configuration, while the outer ring shows the phase-correlation between channels.

Phase correlation between adjacent channels is shown as a white section with a length proportional to the correlation. Additionally, L-R phase correlation is displayed on the top portion of the ring, and L-C and C-R inter-channel phase correlations are displayed just above the top of the ring.

The physical locations of the speakers for the selected configuration are marked on the ring itself for reference.

To display meaningful information, Nebula surround should be informed about the current speaker layout you are using. The speaker layout can be selected inside the IO menu.



19.2 Settings

19.2.1 Mode

Name	Description
View	Choose between the top or front view.
LFE	Activate the influence of the LFE in the Nebula computation
Floor phase	Display the phase relationship of the floor speakers
Overhead phase	Display the phase relationship of the overhead speakers
FloorOverhead phase	Display the phase relationship between the floor and the overhead speakers
Hide phase > 0	Hide phase values over 0
Speakers	Display the speaker layout
Head	Display the listener head
Axes	Display the polar grid

19.2.2 Scale

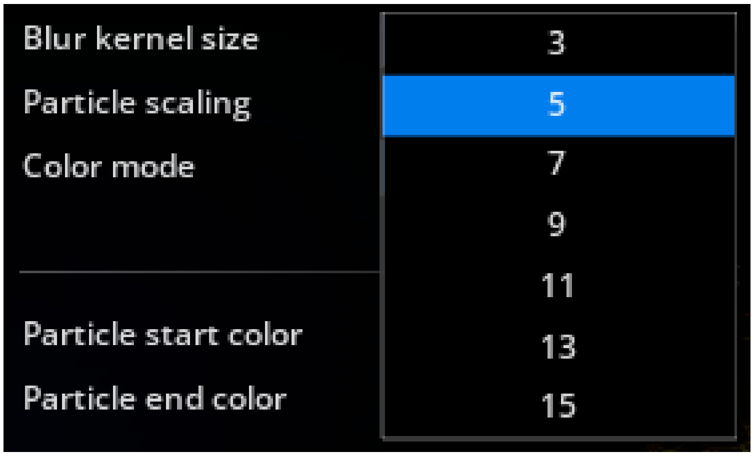
Name	Description
Focus	Controls the stereo image width X-axis display range, in dB. A value comprised between ±18 and ±24dB correlates well with our ability to perceive the stereo image. Default is ±18dB.[1]
AutoScale	This parameter controls whether the overall audio level variations modulate the intensity of the particles. In essence, when enabled, the color nuances will vary according to the relative amplitude of a frequency, allowing to monitor the relative amplitude spectrum variations. When disabled, the color will reflect the absolute audio level. You can also think of this as a kind of auto-gain setting.
AutoScale release	This controls whether color variations should be smoothed in time or not. When engaged, color variations is slowed down a bit, which makes overall level transitions more obvious.[2]
Linear blend range	Adds a constant blend amount to the particle. This ensures some particles are always blended into the image even if there magnitude is low. A low value for this setting stabilizes the appearance of particles. With large values more of the spectrum dynamics are taken into account, and only peaks mostly come through.
Log blending	Toggles between linear and logarithmic blending of the current particle with old particles. The default is off, i.e. linear blending, which tends to favor the display of peaks. Logarithmic blending, on the other hand, preserves more of the full dynamic range of the data, and also gives some visibility to lower levels.

19.2.3 Display

Name	Description
Passes	Determine the number of drawing passes to create the particle clouds on screen. Lower values will emphasize individual particles, while greater values create zones. Such zones can help in reading the information provided by nebula.
Fading	Controls display persistence, i.e. the “fade to black” amount for a frame. Lowering this value retains past particles longer, whereas increasing this makes them disappear faster.

[1] Pixels outside the focus range are clamped to the view boundaries.

[2] You should enable this setting when you want to visualize quick level variations such as those that frequently occur in movie soundtracks.

Name	Description
Size factor	Controls the size of individual particles with respect to screen size.
Blur kernel size	 Controls the radius of the blur effect applied to past particles. Particles are “smeared” more and more as they become older, depending on this setting. Naturally, a bigger value increases the smearing, at the expense of processing power.[3]
Particle scaling	Toggles automatic adjustment of particle size with screen size. When enabled, the overall aspect of the display will remain similar even if the view size changes.
Color mode	Provides the following particle-coloring modes: - Power: the color varies according to the power of the signal in the frequency region - Dynamics: same as previous except this mode works on signal dynamics - Power / dynamics: a mix of the above - Frequency: the color varies according to frequency only, using a rainbow-palette

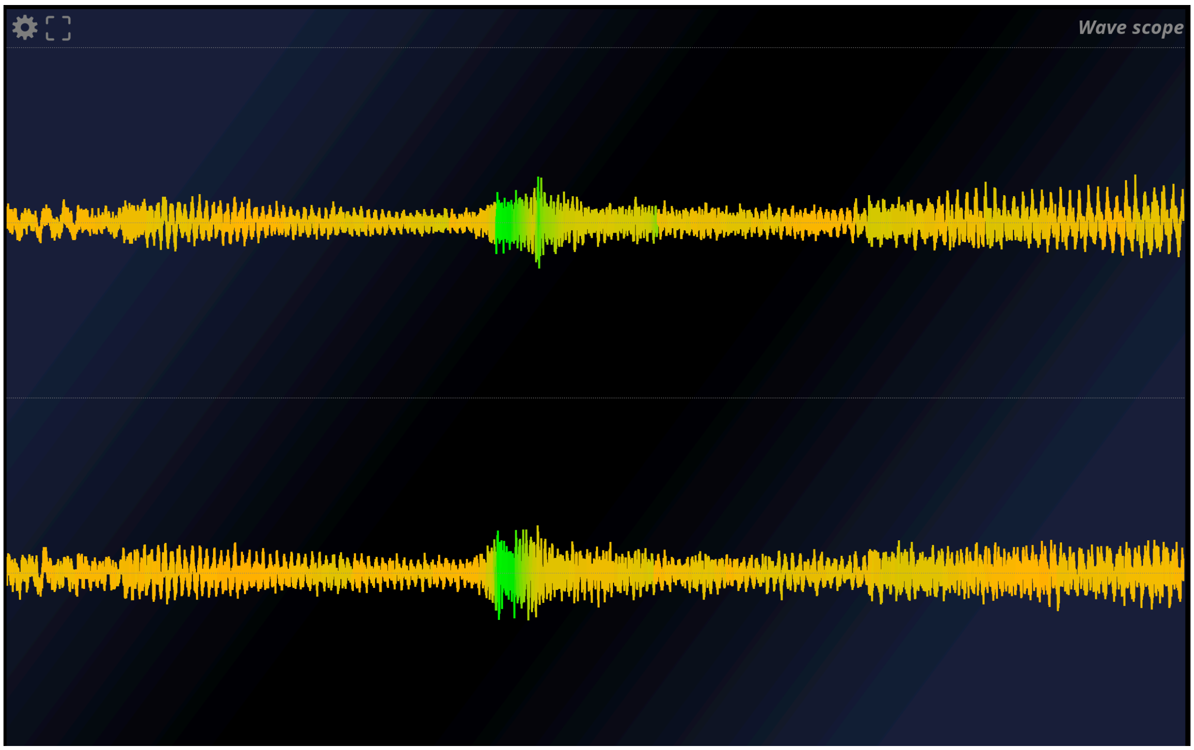
19.2.4 Power color grading

Name	Description
Particle start color	Sets the color to use for maximum amplitude when color mode is set to Duotone.
Particle end color	Sets the color to use for minimum amplitude when color mode is set to Duotone.

[3] Choosing the value for this setting is really a matter of taste, although please keep in mind that values above 5 will require a sufficiently powerful graphics card in order to maintain a responsive display.

20 Wave scope

The wave scope is a simple oscilloscope-type waveform display.



Wave scope display with stereo input.

20.1 Settings

20.1.1 Setup

Figure 20.1: Wave scope setup options

Name	Description
Time	Time window in milliseconds.
Color Mode	 **Static**: Displays the waves using 1 unique static color. **Custom Dynamic**: Displays the waves according to the transient using a 2 user defined color gradient.[1] **Rainbow Dynamic**: Displays the waves according to the transient using a rainbow color gradient.

[1] If “custom dynamic” is chosen, user defined “Color 1” and “Color 2” will be used.

21 RMS metering

21.1 About Metering

All meters display the current signal meter values as solid vertical bars, and the peaks are indicated with horizontal lines at the corresponding value. Peak hold time can be adjusted in the settings if necessary. The peak value is also displayed in a numeric format at the top of the meter, which is emphasized in red in case of clipping or overload.

Several meter displays are available, each scrupulously implementing one of the more common and up-to-date industry norms, as detailed in the following paragraphs.

21.2 Introduction

RMS, which stands for Root Mean Square, is a measure of the average magnitude of a varying signal, or equivalently, the average power over the signal over a time period, called the integration time.

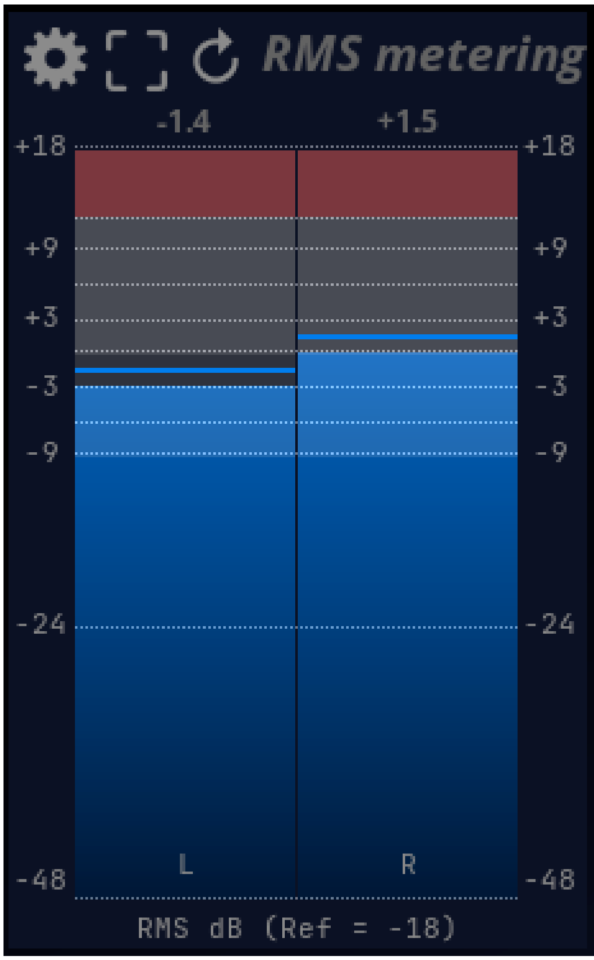
> **Note**
>
> The live layouts display dB SPL (Sound Pressure Level) values, which is the standard measure of acoustic pressure. This requires that your input chain first be calibrated in order to get accurate and meaningful readings, as factors such as your particular microphone's sensitivity and preamplifier gain are not known in advance. For this, you will need to get your hands on a calibrator, which is a box fitted with a transducer that outputs a known acoustic level and features a socket designed to hold the microphone.

21.3 Preset

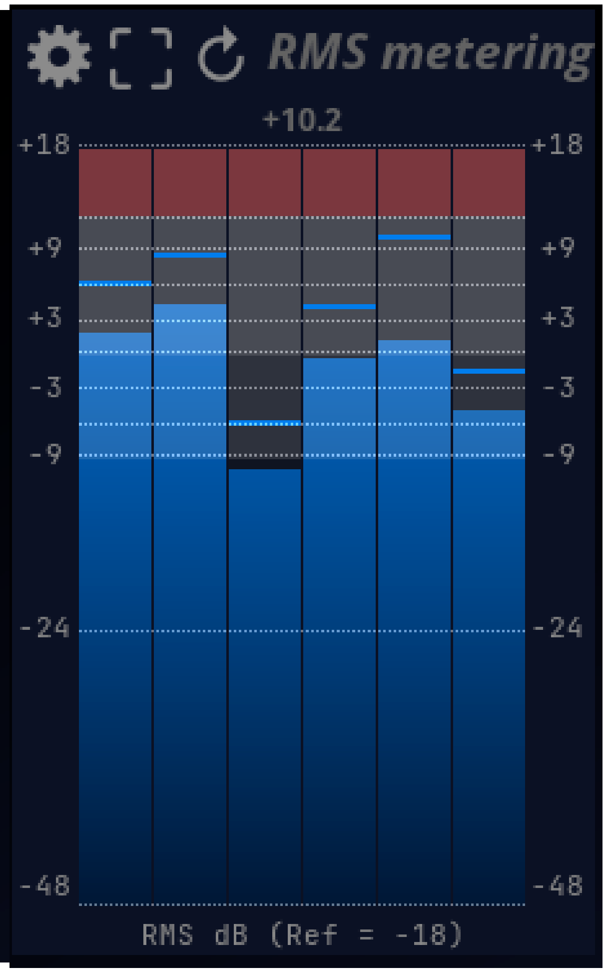
A number of presets covering widely and not so widely-used metering standards are provided.

Name	Description
Custom	User defined values.

Name	Description
Default	All-round settings with: - From -48 to +18 dB range, referenced at -18dB. - 160ms integration time, 16dB/s release, 1dB peak release and 60 frames peak hold.
Ref -18dB A/B/C/K	Default settings with pre-equalization following either normalized ANSI A/B/C or ITU-R BS.1170-2 weighting curves, referenced to -18dB.
Ref -20dB A/B/C/K	Default settings with pre-equalization following either normalized ANSI A/B/C or ITU-R BS.1170-2 weighting curves, referenced to -20dB.
VU meter	Standard reference VU settings, with 300ms integration, 66/7dB/s release and peak release times, referenced at 0VU/-4dBu/-18dBFS.
Standard	The scale is non-linear and covers -20 to +3VU, complying with IEC 60268-17.
K-System / VU	Linear scale, conforming to Bob Katz's recommendations, referenced at either -12, -14 or -20dB, 300ms integration, 66.7dB/s release and 12dB/s peak release times, 180 frames peak hold.
K-System / Slow	Identical to K-System/VU, except that integration times are doubled. This reflects Bob Katz's view that Vu-meter timings are appropriate for speech, but that longer timings are better suited to music.
DIN 45406	This preset conforms to the standard used many European broadcasters such as French (PAD) and German (IRT) television. Integration time is 10ms for a 90% signal increase; fall-back time is 1.7s per 20dB; with a linear scale covering a range from -50 to +5dB, referenced at -9dBFS. The corresponding standards are DIN 45406, IEC 60268-1, and ARD Pfl.H.3/6.
Nordic N9	5ms integration time for an 80% increase, fall-back time 1.7s per 20dB, linear scale covering the range from -40 to +9dB, referenced at -18dBFS, according to IEC 60268-10/1 + N9 supp.
BBC Normal	10ms integration time for an 80% increase, fall-back time 2.8s per 24dB, custom scale with graduations spaced apart by 4dB, and 4 stands for the -18dBFS reference, according to IEC 60268-10/2a.
BBC Slow	Same as above except for ballistics, where the integration time is changed to 69.2ms for an 80% increase, and 3.8s per 24dB fall-back.
EBU Normal	10ms integration time for an 80% increase, fall-back time 2.8s per 24dB, linear scale covering the range from -12 to +12dB, referenced at -18dBFS, according to IEC 60268-10/2b.
EBU Slow	Same as above except for ballistics, where the integration time is changed to 69.2ms for an 80% increase, and 3.8s per 24dB fall-back.



(a) RMS meters with stereo input.



(a) RMS meters with 5.1 surround input.

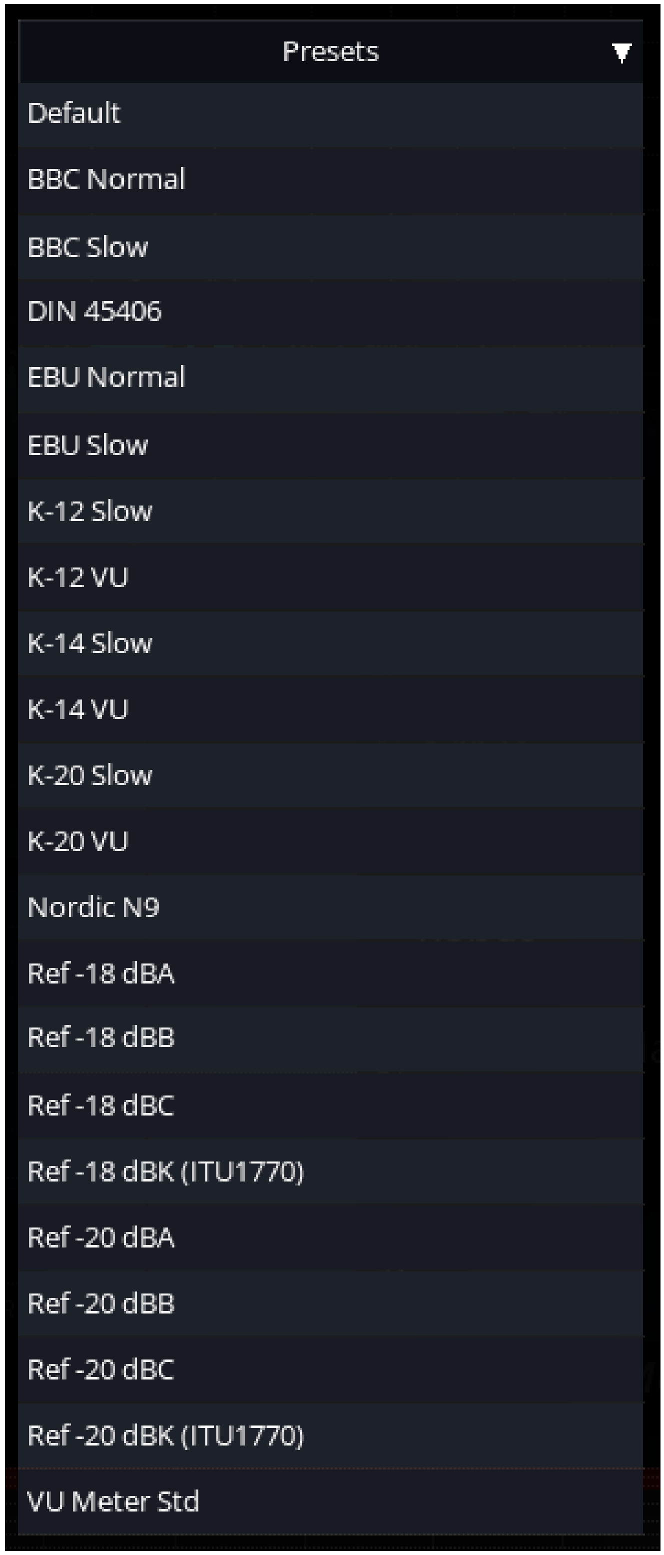


Figure 21.3: Available RMS metering presets

21.4 Settings

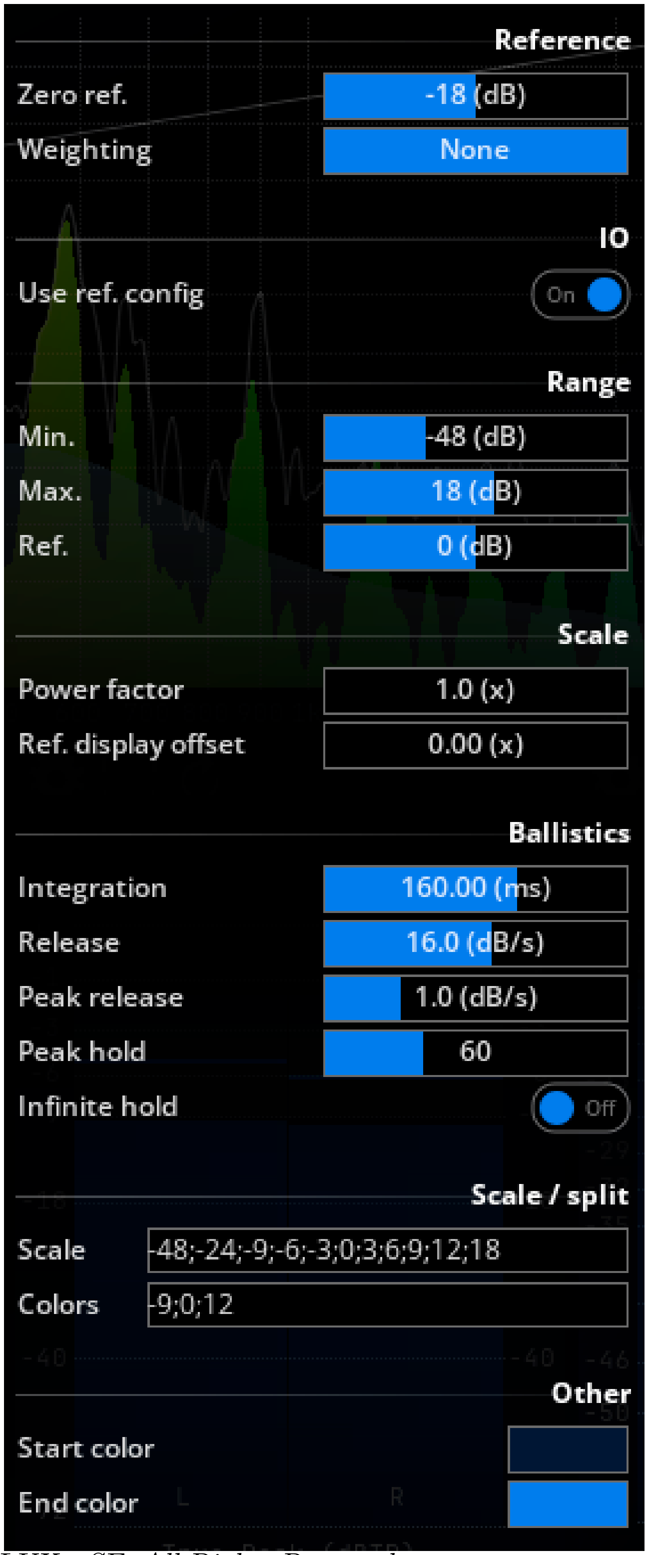
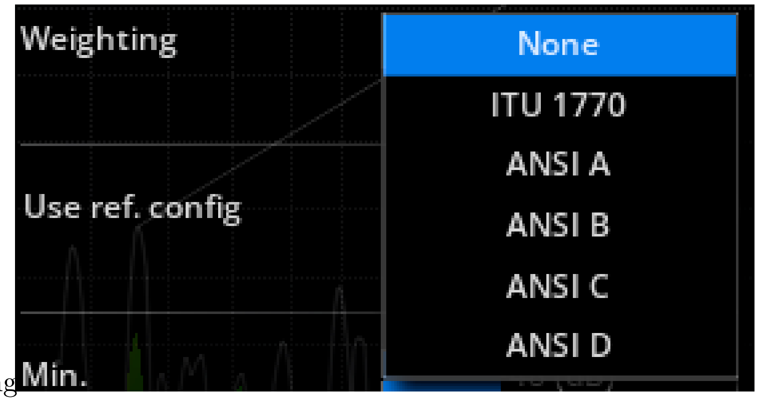


Figure 21.4: RMS metering setup options

21.4.1 Reference

Name	Description
Zero reference	Adjusts the reference point. Default is -18dB (DVD standard). Do not change this unless you specifically want to divert from the standard, as this will otherwise compromise meter readings. Standard values are -18dB for DVD authoring and -20dB for film.
Weighting	 Applies an optional weighting filter conforming to various standard curves: - None (default). - ITU 1770: K-weighting filter, comprising a shelving and a high-pass (RLB-weighting) filter in series, as specified in ITU-R BS.1170-2 and employed by EBU R128 (PLOUD). - ANSI A, which is roughly the inverse of the Fletcher-Munson curve. - ANSI B. - ANSI C. - ANSI D.

21.4.2 IO

Name	Description
Use input (reference) layout	Define if the number of channels displayed by the meter reflects the current input reference layout or the number of channels of the system tuning inputs.

21.4.3 Range

Name	Description
Min / max	Defines the minimum and maximum values to be displayed on the meter bars. This does not affect the text readings above the bars.
Ref.	Defines the reference value for the scaling offset.

21.4.4 Ballistics

Name	Description
Power factor	Apply a scaling factor to the display. Greater values increase the precision around 0 dB.
Ref. display offset	Offset the display reference. Greater values emphasize RMS levels above 0 dB.

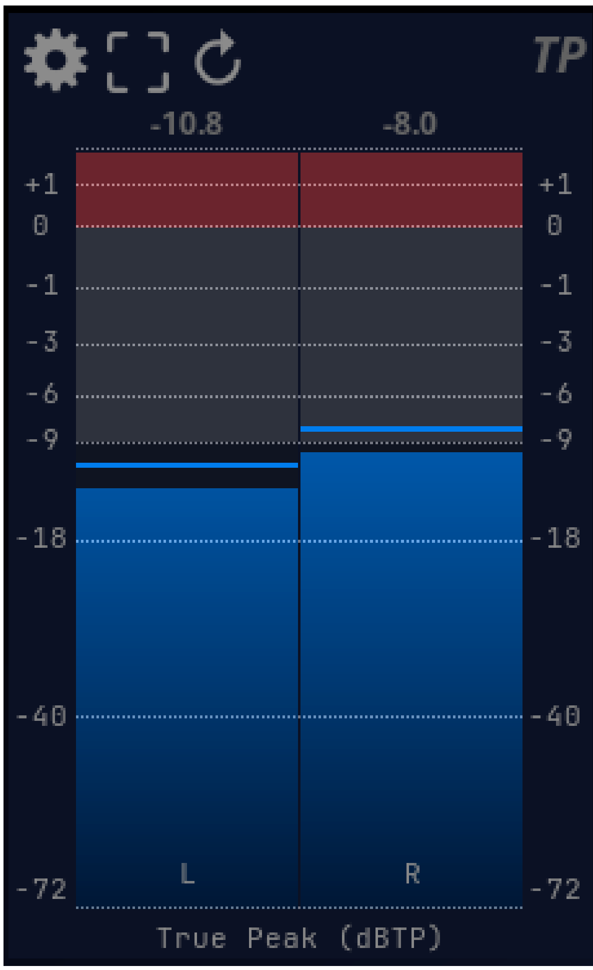
21.4.5 Scale

Name	Description
Integration	Defines the meter integration time constant, in milliseconds. This corresponds to the length of the time window over which an RMS level value is computed. Decrease this to respond to signal level variations more quickly, at the expense of meter precision and stability. Default is 160ms.
Release	Release time of the meter, in decibels per second. This controls the falloff rate of the meter. Decrease this to respond to signal level variations more quickly, at the expense of readability. Default is 16 dB/s.
Peak re-lease	Release time of the peak indicator, in decibels per second. This controls the falloff rate of the peak hold indicators. Increase this to retain peaks for a longer time. Default is 1dB/second.
Peak hold	Sets the number of display frames to wait until the peaks actually start to fall-back to zero. Default is 60 frames.
Infinite hold	Hold the maximum peak value registered forever.
Scale & split	Define the levels where the meter show different lines and colors.

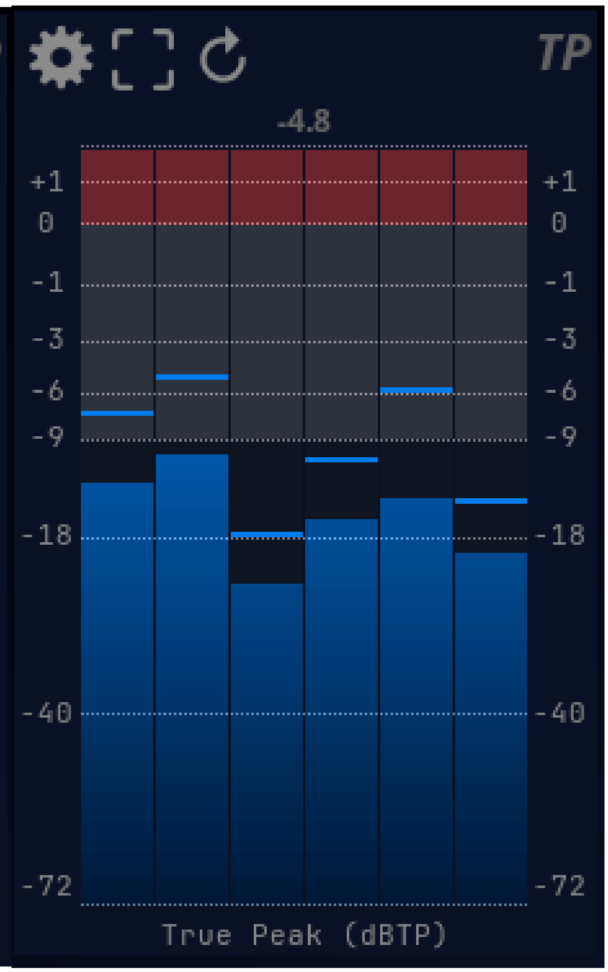
21.4.6 Other

Name	Description
Start color	Define the bottom color of the gradient used to draw the bar graph.
End color	Define the top color of the gradient used to draw the bar graph.

22 True peak metering



(a) True-peak meter with stereo input.



(a) True-peak meter with 5.1 surround input.

All digital audio wave signals are ultimately converted back to analog at some point, and while it is often desirable to maximize the overall volume of a signal or a complete mix, care must be taken in order not to go above the digital scale zero decibel ceiling, or nasty distortion and clipping will occur. This common and widely used rule is, however, not entirely sufficient, as the digital and analog processing involved in a D/A converter does not guarantee that a 0 dBfs peak signal will exactly translate to a 0dB peak in the analog domain.

Without getting into too much detail, this phenomenon can be attributed to the oversampling and reconstruction filters present in the D/A convertors, whose roles are to rebuild a continuous-time signal from a set of discrete digital values sampled at regularly spaced time intervals. This interpolation process can therefore generate values which lie above 0dB, which is known as overshoot.

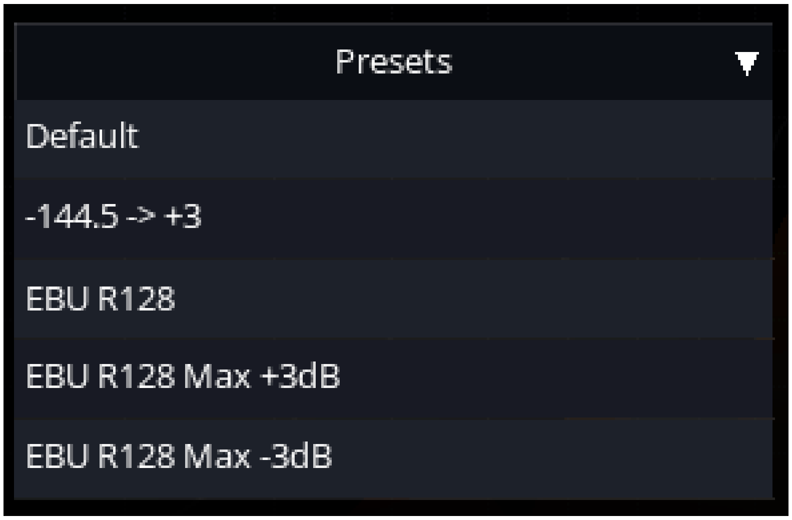
Relying solely on the peak value of samples can lead to the following problems:

- Inconsistent readings between successive playbacks of the same material.
- Unexpected overloads of the D/A output converter.
- Under-readings and beating of pure tones.

TruePeak metering aims to overcome these limitations by mimicking parts of the D/A conversion process, effectively up-sampling the measured signal, in order to display the true value of peaks that occur in the analog domain.

22.1 Preset

Name	Description
Custom	User-defined values.
Default	This preset uses the following all-round settings: - Range: -72 … +3 dB referenced at 0dB. - Scale: 1.8x power factor, 0.06x reference display offset. - Ballistics: 16dB/s release time, 1dB/s peak release, 60 frames peak hold. - Scale / split: -72, -40, -18, -9, -6, -1, 0, +1, +3 dB.
EBU R128	Referenced at -1dB.
EBU R128 Max -3dB	Referenced at -3dB.
-48.0 -> +3	Limited -48 … +3dB range with adapted scale/split values.
-144.5 -> +3	Wide -144.5 … +3dB range with adapted scale/split values, to monitor the full 24-bit dynamic range and possible clipping.



22.2 Settings

22.2.1 IO

Name	Description
Use input (reference) layout	Define if the number of channels displayed by the meter reflects the current input reference layout or the number of channels of the system tuning inputs.

22.2.2 Range

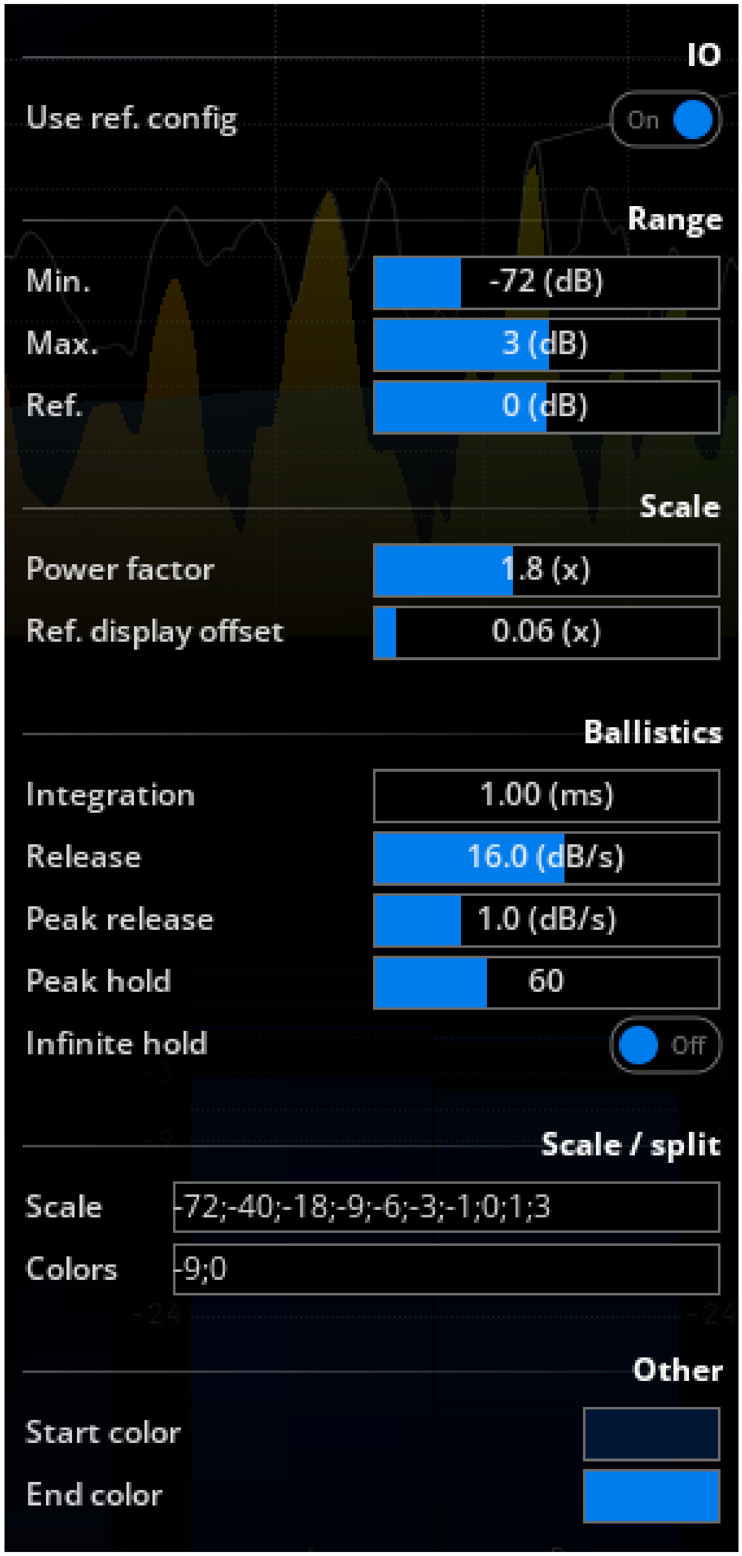


Figure 22.3: TruePeak metering setup options

Name	Description
Min / max	Defines the minimum and maximum values to be displayed on the meter bars. This does not affect the text readings above the bars.
Ref	Controls the position of the reference value on the display. This does not affect the meter values per se; it is a cosmetic setting only.

22.2.3 Scale

Name	Description
Power factor	Controls the scaling of the display with respect to meter values. This allows to stretch or compress the display around Reference.
Ref pixel offset factor	Adjusts the offset of the reference value (Reference) with respect to meter height.

22.2.4 Time

Name	Description
Release	Release speed of the meter in decibels per second.
Peak release	Release speed of the peaks in decibels per second.
Peak hold	Number of frames to hold the peaks before the actual release phase begins. Sixty frames correspond to 1 second on a fast system, capable of a 60Hz refresh rate.
Infinite hold	When enabled, peaks are held until the next reset, which is useful for checking a whole track never clips.
Reset	 Button resets the meter to its initial state (values and peaks at minimum).

22.2.5 Scale & split

Scale

Name	Description
Scale	Meter labels are defined here as a comma-separated list of dB values to be shown on the side of the meters. This also defines where the corresponding horizontal markings are. Default is -72;-40;-18;-9;-6;-3;-1;0;1;3.
Colors	This lets you customize the values at which color transitions occur. You can enter as many values as you wish, as a comma-separated list, but make sure the values are in increasing order. The last value always defines the clip level, which will be indicated in red. Default is -9;0.
Other	Controls whether meters are drawn with texture or in a plain solid color. Default is on.

22.2.6 Other

Name	Description
Start color	Define the bottom color of the gradient used to draw the bar graph.
End color	Define the top color of the gradient used to draw the bar graph.

23 Loudness metering

23.1 Loudness ITU-R BS 1770 &amp EBU R128 PLOUD

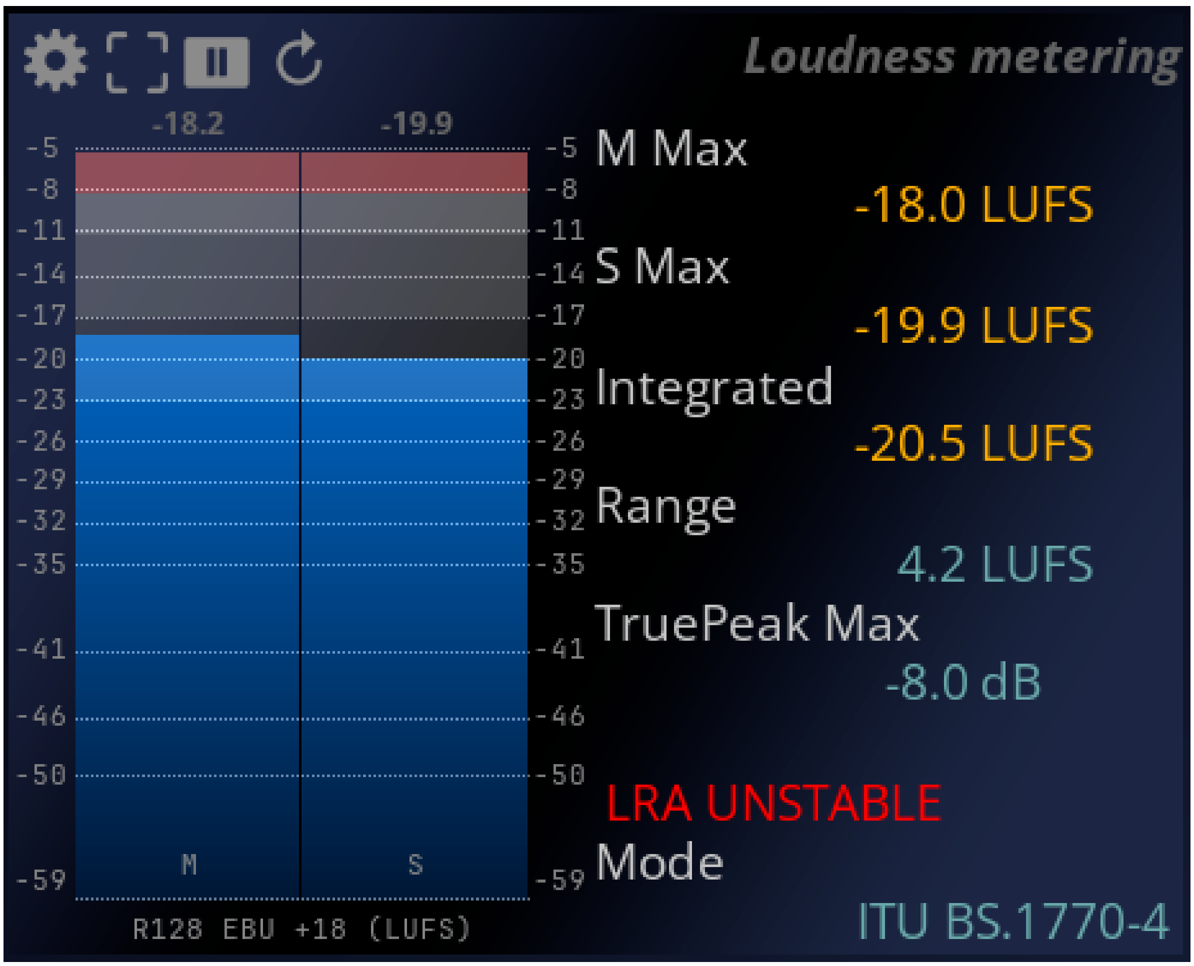


Figure 23.1: Loudness meter.

ITU-R BS.1170-4 and EBU R128 recommendations introduce a new paradigm for audio metering, which define a way to measure the perceived loudness of audio material in a normalized and reproducible manner.

Please refer to the official documents freely available online at tech.ebu.ch/groups/ploud * or consult a reference book such as "Audio Metering. Measurements, standards and practice" by Eddy Brixen (Focal Press, ISBN 9780240814674) for detailed information on this subject.

23.2 Principles

23.2.1 Units

ITU-R BS.1170-2 notably defines LU (Loudness Unit) and LUFS (Loudness Unit, referenced to Full Scale) units, which are used by EBU R128, and Maximum True Peak Level.

- LU is used for measurements *relative* to a reference level and measuring range.
- LUFS is used for *absolute* measurements.

The meter display is switchable between LUFS (absolute, default) and LU (relative). The target loudness level to aim for is -23 LUFS = 0 LU.

> **Note**
>
> Some documentation refers to the LKFS unit. LUFS and LKFS are synonymous, but LUFS should be used as it conforms to scientific unit notation.

23.2.2 Loudness and EBU mode

EBU mode specifies three time scales corresponding to three different, complementary loudness levels

- M: Momentary, 400ms integration time
- S: Short-term, 3s integration time
- I: Integrated from start of measurement or last reset, gated

> **Note**
>
> Loudness is a measure of global loudness, so individual channel metering is irrelevant in this context.
> No additional slowdown of the attack or release of the meter is employed, as indicated by the norm.

The integrated loudness can be understood as the overload loudness of the audio over time, excluding very soft passages through the use of absolute and relative gating.

23.2.3 Loudness Range (LRA)

Loudness range measures the average long-term variations of the loudness; it is expressed in LU.

23.2.4 Scales

EBU R128 specifies two normalized scales:

- EBU +9, ranging from -18.0 LU to +9.0 LU (-41.0 LUFS to -14.0 LUFS)
- EBU +18, ranging from -36.0 LU to +18.0 LU (-59.0 LUFS to -5.0 LUFS) (Default)

23.3 Dolby Dialogue Intelligence

23.3.1 Introduction

While EBU R128 aims to measure global perceived loudness, irrespectively of the audio material, Dolby Dialogue Intelligence is a patented technology designed to specifically measure the perceived loudness of dialogue elements in the audio. It is therefore targeted towards broadcast applications.

23.3.2 General principle

Dialogue Intelligence replaces EBU R128's level-based gate with a speech-content ratio based gate. The algorithm computes several low-level features for the incoming signal in speech channels. These are then combined into an overall speech percentage figure. When speech content is detected, Integrated Loudness is computed from the speech channels which have a speech content ratio above a certain threshold.

When another material is detected, i.e. not speech, standard EBU R128 Integrated Loudness computation is employed.

23.3.3 Display

The current speech content is displayed as text below the current gate status.

Additionally, color coding indicates the speech content ratio.

- **Speech** : speech content present
- **Green**: high speech content
- **Orange**: medium speech content

- **Red**: low speech content
- **Other**: other material present

23.3.4 Delay and compensation

The sophistication of the algorithms employed in Dialogue Intelligence incurs an overall latency of 2048ms (approx. 2s).

When Dialogue Intelligence is enabled, the display of other Loudness values is compensated to make sure meter readings are consistent. Other real-time meter (RMS, TruePeak) displays are not compensated, as we feel, in this case, maintaining the best reactivity to the incoming signal is more important.

All meter statistics are time-aligned.

23.3.5 Surround

Channels taken into account by the algorithm are determined based on the current channel configuration.

For mono/stereo signals, all channels are taken into account. For surround configurations, only Left/Right and Center channels are considered, if present.

> ℹ Note
>
> Dialogue Intelligence computation only affects I (Integrated) Loudness values.
> Toggling Dialogue Intelligence on and off forces a reset of the meter values.

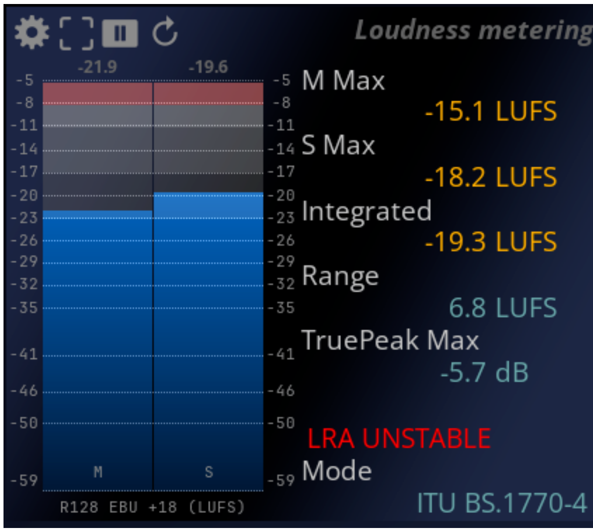
23.4 Controls and display

23.4.1 Display

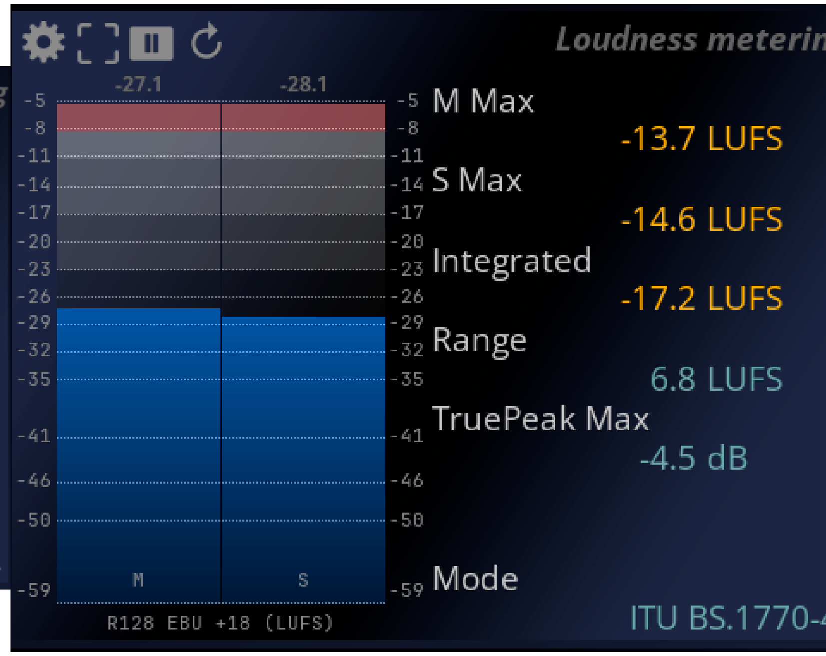
The meter display has the following arrangement:

- left bar: Momentary Loudness value
- right bar: Short-term Loudness
- text overlay: Integrated Loudness and Loudness Range (LU) values, Gated indicator lights red when the gate is active

The Loudness Range value is displayed once the measurement has been running for at least 60 seconds, according to the EBU Tech 3342 specification, otherwise a 'LRA Unstable' warning is shown.



(a) Unstable LRA



(a) Stable LRA

23.4.2 Pause

Clicking the button pauses measurement; clicking again resumes it. This allows you to make adjustments without affecting Integrated Loudness, instead of having to start all over again.

23.4.3 Reset

Clicking the button resets the meter to its initial state.

> **Note**
>
> Don't forget to reset the Loudness meter if you're starting playback of a new track, as Integrated Loudness, by design, measures the overall Loudness since the last reset.

> Otherwise, you'd be measuring the overall Loudness of the combined tracks, which is probably not what you want.

23.5 Settings

23.5.1 Mode

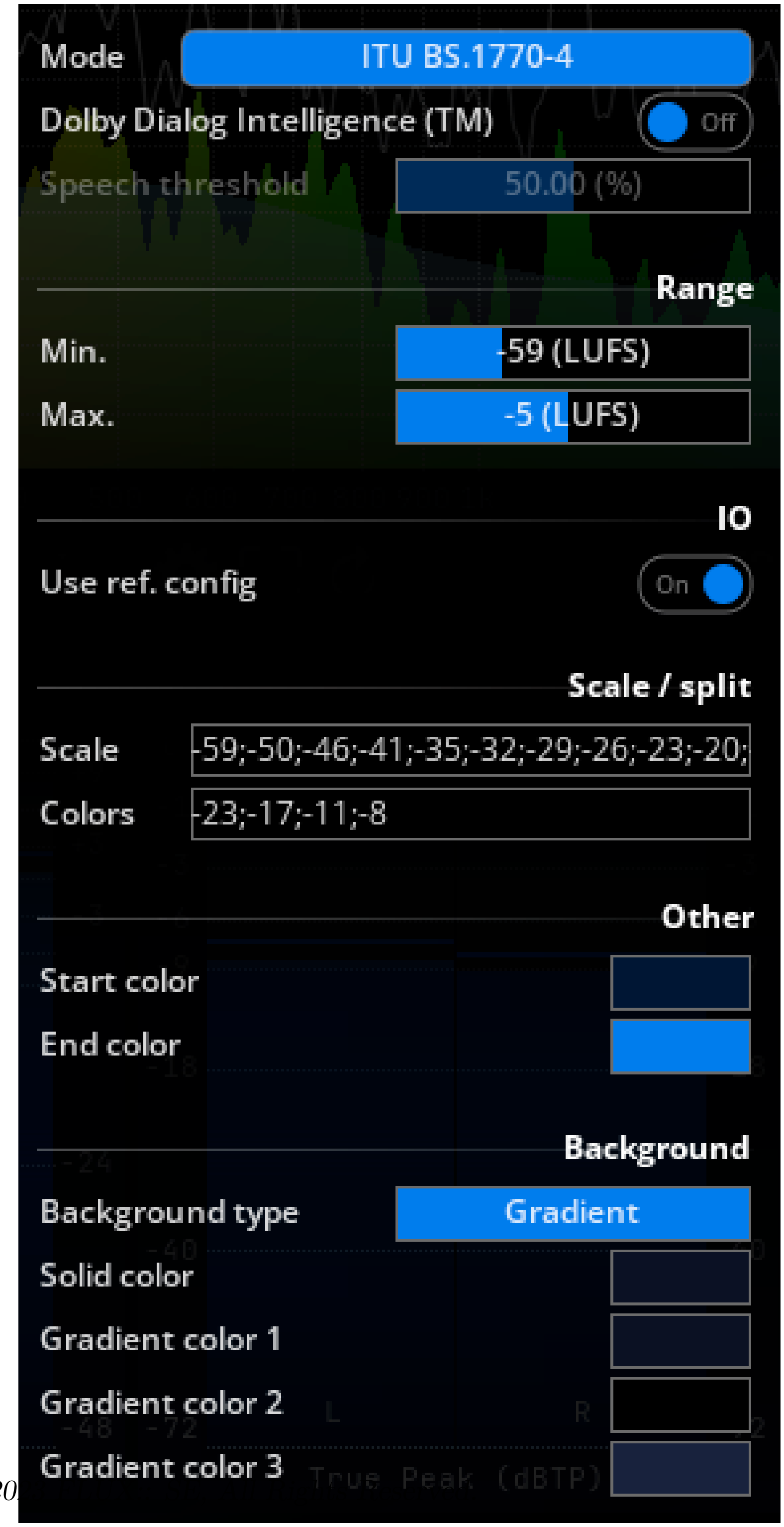


Figure 23.4: EBU R128 Loudness metering setup.

Change the targeted integrated loudness as well as the maximum True Peak level.

Name	Description
ITU BS.1770-1	Defined as a maximum integrated level of -24 LUFS and a maximum True Peak level of -2 dBFS. You can read the official publication here.
ITU BS.1770-2	Defined as a maximum integrated level of -24 LUFS and a maximum True Peak level of -2 dBFS. You can read the official publication here.
ITU BS.1770-3	Defined as a maximum integrated level of -24 LUFS and a maximum True Peak level of -2 dBFS. You can read the official publication here.
ITU BS.1770-4	Defined as a maximum integrated level of -24 LUFS and a maximum True Peak level of -2 dBFS. You can read the official publication here.
EBU R128	The EBU R128 is European TV Broadcasting standard. It is defined as a maximum integrated level of -23 LUFS with a tolerance of 0.5 LUFS and a maximum True Peak level of -1 dBFS. You can read the official publication here.
EBU R128 S1	The EBU R128 S1 is a supplement of the EBU R128 for short-form content like advertisements (commercials) and promos (as well as interstitials, etc.). For such programs, there is a need to give guidance using maximum short-term loudness in addition to the integrated loudness and the maximum true peak level. It defines a maximum integrated level of -23 LUFS, a maximum short-term loudness of -18 LUFS and a maximum True Peak level of -1 dBFS. You can read the official publication here.
ARIB TR-B32	The TR-B32 recommendation from the Association of Radio Industries and Businesses (ARIB) in Japan provides guidelines for broadcasting, focusing on loudness management to ensure consistent and high-quality audio. It specifies a target loudness level of -24 LUFS and a true peak limit of -1 dBTP to prevent distortion. Measurements should comply with the ITU-R BS.1770 standard. You can read the official publication here.
ATSC A/85 (2011)	The A/85 recommendation from the Advanced Television Systems Committee (ATSC), published in 2011, provides guidelines for audio loudness management in broadcasting to ensure a consistent and high-quality listening experience. It specifies a target loudness level of -24 LUFS and a true peak limit of -2 dBTP (decibels True Peak) to prevent distortion. Measurements should comply with the ITU-R BS.1770 standard.
ATSC A/85 (2013)	The 2013 revision of the A/85 recommendation from the Advanced Television Systems Committee (ATSC) builds upon the 2011 guidelines to further refine and clarify loudness management practices in broadcasting. While keeping the same target levels, it now refers to compliance toward the third revision of the ITU-BS.1770. You can read the official publication here.

Name	Description
Free TV OP-59	The OP-59 recommendation from Free TV, Australia, provides guidelines for the measurement and management of loudness in television broadcasting. It specifies a target loudness level of -24 LUFS and a true peak limit of -2 dBTP. You can read the official publication here.
AGCOM 219/09/CSP	The 219/09/CSP recommendation from AGCOM (Autorità per le Garanzie nelle Comunicazioni), Italia, provides guidelines for the measurement and management of loudness in television broadcasting. It specifies a target loudness level of -24 LUFS and a true peak limit of -2 dBTP. You can read the official publication here.
Portaria 354	The Portaria 354 is a standard defined by the Ministry of Communications, Brazil based on ITU-R BS.1770-2 and EBU R128 (2011) with the integrated target level of -23 LUFS and a target true peak of -2 dBFS. You can read the official publication here.
Sony R001 HOME	For home-based SCE platforms, the average loudness level of audio content should be normalized to a target of -24 ($\pm$2) LUFS. Additionally, the maximum true peak level should not exceed -1 dBTP, using a meter compliant with both ITU-R BS.1770-3 and EBU Tech Doc 3341. You can read the official publication here.
Sony R001 PORTABLE	Analogous to the home version, but the average loudness level of audio content should be normalized to a target of -18 ($\pm$2) LUFS. You can read the official publication here.
AES Streaming	The AES Technical Document TD1004.1.15-10 recommends that the Target Loudness of audio streams should not exceed -16 LUFS to prevent excessive peak limiting and should not be lower than -20 LUFS to ensure audibility on mobile devices. Short-form programming, such as commercials, should have a Maximum Short-term Loudness of no more than 5 LU higher than the Target Loudness. Additionally, the maximum peak level should not exceed -1.0 dB TP. The metering should conform to the EBU R-128. You can read the official publication here.
Spotify	Spotify is a music streaming service which recommends a normalization of the integrated level at -14 LUFS and limiting the true peaks at -1 dBTP. Audio contents that don't meet this loudness requirement will be attenuated if too loud and amplified if too quiet. Not that no limiting is applied if the audio is too low and the algorithm preserves a headroom of 1 dBTP. You can read the official publication here.
Spotify Loud	The loud mode in Spotify app applies a limiter to normalize the integrated level of audio content up to -11 LUFS while leaving the maximum true peak at -2 dBTP. Our metering reflects on these two values. You can read the official publication here.
YouTube	Youtube is a video hosting web site which provides a similar recommendation to Spotify by normalizing the content to -14 LUFS-I and by leaving the maximum true peak at -1 dBTP.

Name	Description
Apple Music & Podcast	Apple Music is a music streaming service which provides a different set of recommendations: - A targeted loudness of -16 LUFS with a tolerance of 2 LU - A maximum true peak of -1 dBTP
TIDAL	TIDAL is a music streaming service. Its recommendations are identical to Spotify.
Amazon Music & Alexa	Amazon, through its music streaming services, recommends a LUFS-I of -14 and a maximum true peak of -2 dBTP.
Deezer	Deezer is a music stream service which recommends -15 LUFS-I and a maximum true peak of -1 dBTP.
Netflix	Netflix is a web service of video on demand. It recommends an integrated loudness target of -27 LUFS with a maximum true peak of -2 dBTP.
AES TD1008	The AES TD1008 is an international streaming service recommendation toward loudness level of audio content. It recommends a target of -16 LUFS-I with a tolerance of 2 LU. The maximum true peak should not exceed -1 dBTP. You can read the official publication here.

23.5.2 Dolby Dialogue Intelligence

Name	Description
Dolby Dialogue Intelligence (TM)	Toggles usage of Dolby Dialogue Intelligence speech gate.
Speech threshold	Defines the speech content threshold in %. Speech channels with a speech content ratio below this value do not participate in the Loudness computation.

23.5.3 Range

Name	Description
Min.	Minimum Loudness to display on the bar-graphs.
Max.	Maximum Loudness to display on the bar-graphs.

23.5.4 IO

Name	Description
Use input (reference) layout	Define if the number of channels displayed by the meter reflects the current input reference layout or the number of channels of the system tuning inputs.

23.5.5 Scale/split

Name	Description
Scale	Meter labels are defined here as a comma separated list of dB values to be shown on the side of the meters. This also defines where the corresponding horizontal markings are. Default is -72;-40;-18;-9;-6;-3;-1;0;1;3.
Colors	This lets you customize the values at which color transitions occur. You can enter as many values as you wish, as a comma separated list, but make sure the values are in increasing order. Default is -9;0. The last value always defines the clip level, which will be indicated in red.

23.5.6 Other

Name	Description
Start color	Define the bottom color of the gradient used to draw the bar graph.
End color	Define the top color of the gradient used to draw the bar graph.

24 Leq Metering

Figure 24.1: LEQ Meter

24.1 Introduction

Leq encompasses a set of sound level meter specifications, which are described in detail in the BS EN 61672-1 European Standard.

FLUX:: MiRA:: implements the following Leq measurements: time-weighted sound level, time-average sound level and sound exposure level.

Frequency weighting is employed for all measurements, A being the standard and default, although other weightings can be specified if necessary.

The Leq module always measures the audio routed through the Mic channel.

24.1.1 Time-weighted sound level

LA is the root-mean-square sound level obtained after exponential time weighting.

Exponential averaging has the effect of progressively ‘forgetting’ past sample values.

The norm specifies two time-weighting constants:

- Fast: 125ms

- Slow: 1s

> **Note**
>
> The corresponding letter symbol is LAF for an A-frequency weighted and F time-weighted sound level, for example.

24.1.2 Time-average sound level

Time-average sound level is basically an RMS meter with frequency weighting applied.

24.1.3 Sound exposure level

This measures the sound exposure equivalent to a 'dose' received for a second.

> **Note**
>
> It is useful for determining the amount of sound pressure to which listeners have been exposed for a certain duration.
> This value naturally increases with time. For a constant source level, this value increases in a logarithmic fashion.

24.2 Logs and log files

Each Leq meter features two special buttons to create log files. Log files are simple markdown files that register the state of the metering every three seconds. The path of storage of this file is defined in the option of the Leq scope (see below).

All the different Leq scopes in the same layout share the same settings for log files. Specifically, they all write their data to the same output file.

24.2.1 Starting a log file

A log file is created as soon as the "play" button over the scope is pressed. The file is created at the location specified in the scope options.

It is named as follows: analyzer_metering_log_YYYYMMDDHHMM, where, in order, Y is Year, M is Month, D is day, H is Hour and M is Minute. For exemple: analyzer_metering_log_202408291015 is a log file created on the 29th of August 2024 at 10:15 AM.

24.2.2 Custom notes

At any point, you can click on the "pencil" button over the scope to enter a specific note in the log file. Like all other data entries, it will be timestamped with the app's current time code.

24.2.3 Log file content

The log file shows the following data as a table:

- The Timecode
- The maximum true peak level
- The true peak level per channel
- The RMS level per channel
- The different Leq scopes of the layout.
- the loudness

Each row of the table corresponds to a specific timecode.

Next, you will find some global data:

- The maximum true peak level
- The maximum true peak level per channel
- The maximum RMS per channel
- The maximum value for each Leq scope
- The maximum loudness

24.3 Settings

Name	Description
Zero ref.	Adjusts the reference point. See RMS for more information.
Name	The name of the meter.
Input	The audio input of the meter.

Name	Description
Weighting	Frequency weighting employed for metering. Can be switched between ANSI standards (A, B, C, D) and none. The default is A.
Average integration	Indicates the time constant for the metering.
Main display	Switches the main measurement display from time-average sound level (the default) to sound exposure level.

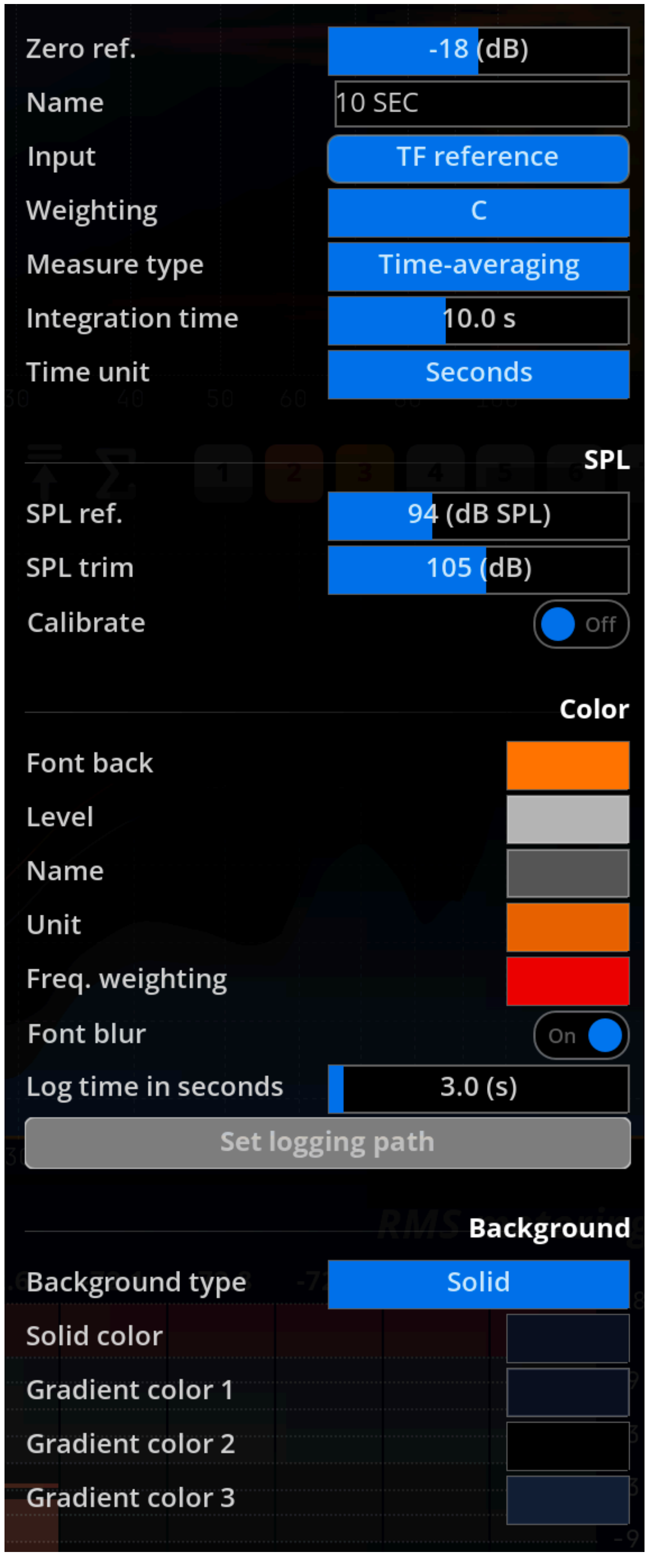


Figure 24.2: Leq settings panel

24.4 SPL

Name	Description
SPL reference	This is the reference level of the calibrator's output, indicated on the device itself or in the corresponding data sheet. A typical value is -94dB.
SPL trim	This is the offset applied to RMS dB values in order to obtain dB SPL readings. It is determined automatically by the calibration procedure.
Calibrate	Press this button after having inserted the microphone into the calibrator socket and activated it in order to determine the SPL trim value.

24.5 Color

The following settings control the visual aspect of the Leq display.

Name	Description
Font back	Common font background color. Main level font color.
Level	Level display color
Name	Name font color
Unit	Unit display font color.
Freq. weighting	Frequency weighting type display font color.
Font blur	Toggles font blurring on (default) and off.

24.5.1 Logs

Name	Description
Log time in seconds	Set the lapse of time between two log entries
Set logging path	Set where the log files are stored.

25 Metering History

25.1 Usage

The metering history panel stores and displays the evolution of meters over time, with a red vertical bar indicating the current time. Start and end time-points of the period are displayed left and right in time-code format.

Selecting which meters are to be included in the display is done by clicking the corresponding buttons in the setup.

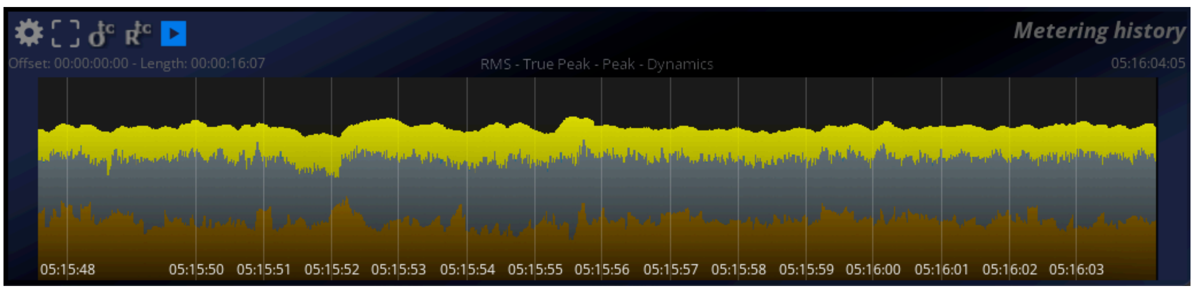


Figure 25.1: Metering history display.

25.1.1 Timecode offset

Clicking the  defines the current time as the Timecode offset.

25.1.2 Timecode offset reset

Clicking the  button resets the Timecode offset to zero. Absolute and relative Timecode will then be the same.

25.1.3 Play

Clicking the toggles history recording on and off. Metering values are discarded when off.

> **Note**
>
> The metering history relies on the same settings as those defined in the various meters. However, when multiple meter values are displayed simultaneously, the display range of the history is adapted to encompass the display ranges of the meters.
> Keep in mind different meters can be set to different zero reference points when comparing meter history curves.

25.2 Settings

25.2.1 TimeCode

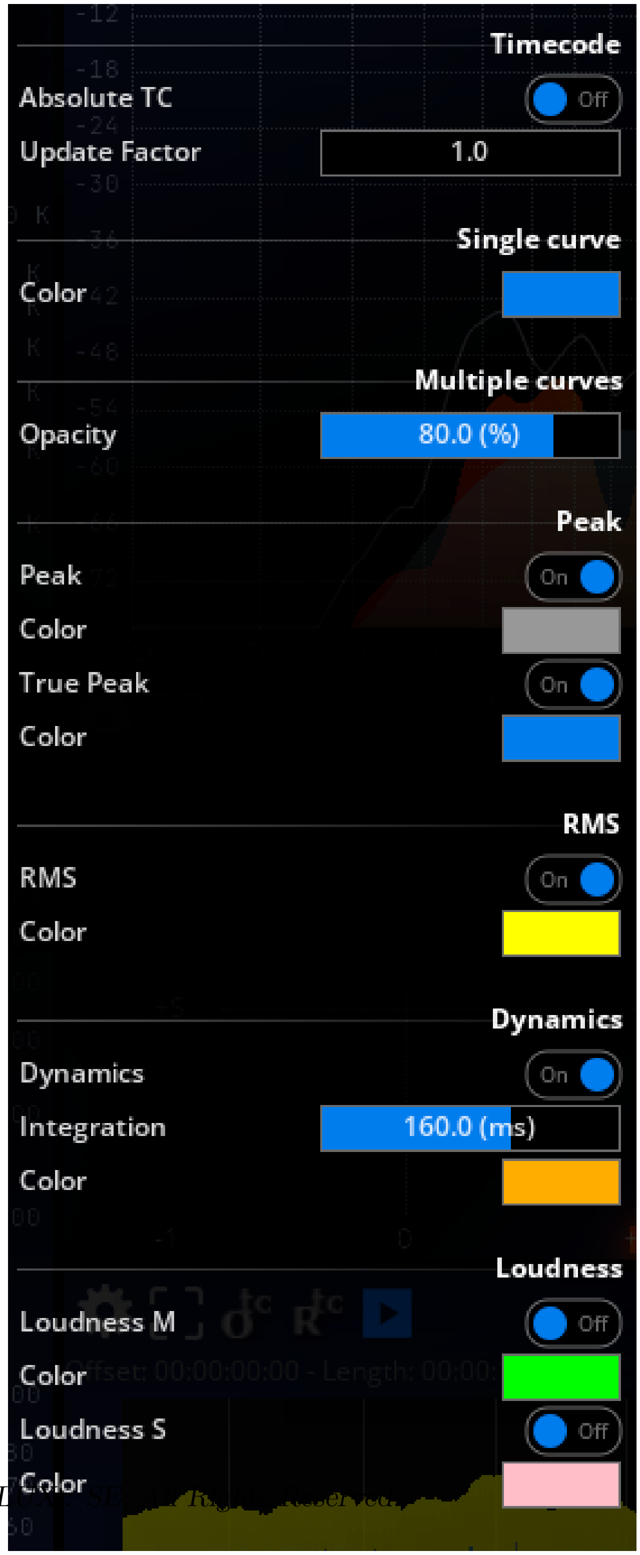


Figure 25.2: Metering history setup options.

Name	Description
Absolute Timecode	Switches between absolute and relative Timecode formats.
Update Factor	Divides the History refresh interval; allowing to increase the history time period.

25.2.2 Single curve

Name	Description
Color	Sets the color to use when only a single curve is selected for display.

25.2.3 Multiple curves

Name	Description
Opacity	Set the amount of opacity for the curves.

25.2.4 Peak

Name	Description
Peak	These settings allow to specify whether Peak and/or TruePeak curves should be displayed, as well the color to use when drawing them.

25.2.5 RMS

Name	Description
RMS	Toggle RMS curve display on and off, and specify the color to use for drawing.

25.2.6 Dynamics

The dynamics is the current dynamic range of the signal, that is the ratio of the peaks with respect to the average, i.e. the crest factor of the signal.

Name	Description
Dynamics	Toggles dynamics curve display on and off.
Integration	Set the integration time, in milliseconds.
Color	Specify the color to use for drawing the curve. [1]

25.2.7 Loudness

Name	Description
Loudness	These settings allow to specify whether Short-term and/or Momentary EBU R128 Loudness curves should be displayed, as well the color to use when drawing them.

[1]Percussive content such as drums or rhythm guitar exhibits high dynamics, as opposed to sustained sounds such as strings and synthesizer pads.

26 Metering statistics

The metering statistics view shows a synthetic view of the current and past meter values in numeric form. It also serves to process multiple existing audio files in one pass, display and export the results to disk.

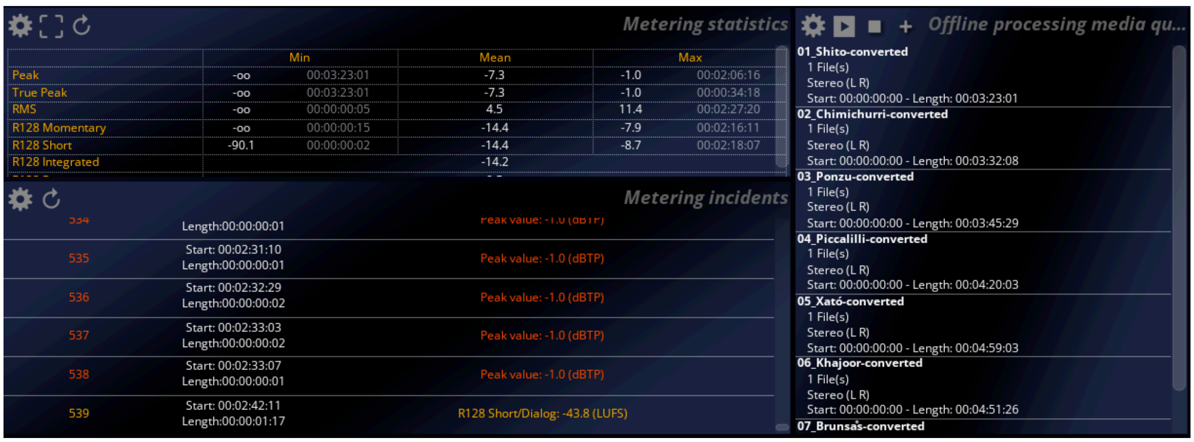


Figure 26.1: Metering statistics display

26.1 Overview

The display shows the average and range for the various level meter values, since the start of the application or the last time the meter was reset, in a spread-sheet type view.

26.1.1 Peak, True Peak and RMS

Mean as well as overall minimum and maximum values are shown. For min. and max. values, the corresponding Timecode position is also indicated.

26.1.2 Loudness

As EBU R128 Loudness already incorporates statistical computations, only the current values are shown.

26.2 File export

Exports a report containing a summary of the metering statistics data to a text file.

Clicking the  button brings up a standard file dialog where you can specify the desired file name for the dialog.

26.3 Setup

26.3.1 Absolute Timecode

Toggles between relative and absolute Timecode display. See TimeCode for more information.

26.4 Incident Reporting

26.4.1 Overview

All TruePeak and R128 Short term values that cross the thresholds are recorded and displayed as a list. Each row in the list shows a record of the offending peak value in dB alongside with the time-code at which the event occurred. You can navigate the list and locate the time positions of the incident, then playback the corresponding source material again in order to identify and correct the problem.

26.4.2 Setup

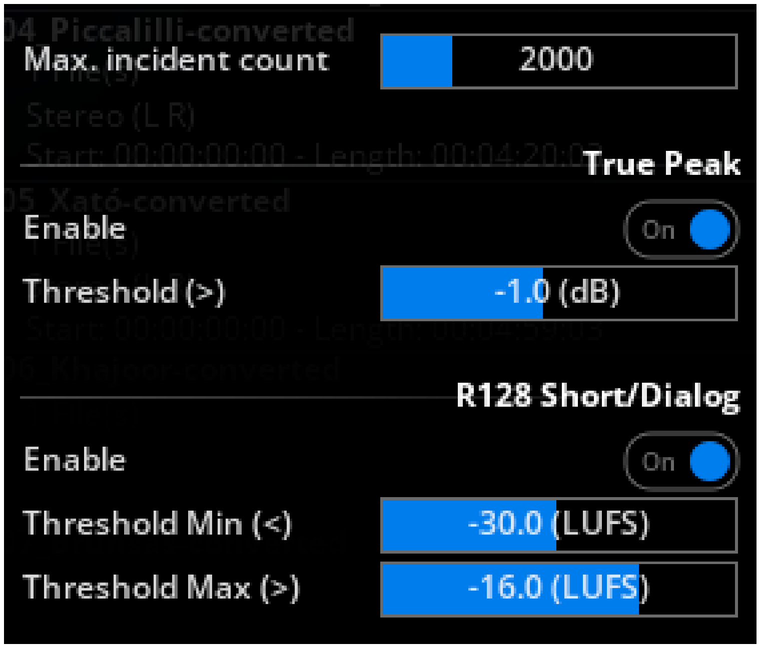


Figure 26.2: Incidents setup options.

Max. incident count

To avoid overloading the display, and eventually, the computer's memory, there is a limit placed on the number of registered incidents, which is 2000 by default. If you go above this, it might be a good idea to back off the master fader a bit anyway to let that music breathe!

However, you can override this behavior by setting this value to -1, which will remove the limit altogether.

TruePeak Incident Enable + threshold

Defines the threshold above which an incident will be registered. Default is 0dBTP, which corresponds to full digital scale. A conservative value would be -0.1dBTP, to be on the safe side.

Keep in mind TruePeak is designed to measure inter-sample peaks, and that 0dBTP is actually a few tenths of decibels softer than digital peak.

EBU R128 Short term / Dialog Incident Enable + thresholds

Defines the threshold under/above which an incident will be registered.

26.5 Off-line Processing Media Queue

26.5.1 Usage

Multiple audio files can be added to the list for unattended queue processing.

Principle

The media queue is intended to process a soundtrack possibly split across several reels and channels. Reels are processed in the order in which they are added and in which they appear in the list.

Usage

Audio files are added by clicking the icon , which brings up a standard file selection dialog, where you can select as many files as you want, provided they all have the same channel count and in a supported format, with a recognized extension (.wav). When you are ready, click the icon to start processing the list, which will be computed much faster than real-time, especially if you have a fast computer.

The results are displayed when ready in the main view, and you can export these to a file just as you would with metering statistics computed on incoming audio.

Reel grouping

If reels are split across several multichannel files, you can add all the files at once directly in the file selection dialog. Reel order corresponds to the order in which the files were added.

Channel grouping

If your source material consists of mono-tracks, you must add reels one-by-one, adding all files for the various channels of the current reel. Please ensure different reels have the same channel count or the software will report an error. Channel configuration and names are inferred from the file names using a fuzzy-logic algorithm that looks for the presence of typical marker characters such as C / Center for the center channel, R / Right for the right channel etc. (case insensitive).

If the automatic channel grouping does not succeed, an error message will be displayed. Please rename the offending file(s) according to one of the expected schemes above to correct the problem.

> **Note**
>
> This function is not intended to process unrelated soundtracks in batch mode.
> Please repeat the operation as necessary if you wish to obtain separate results for individual tracks.

Part VII

System Measurement

Introduction

This user guide section is entirely dedicated to the “System Tuning” layouts of the MiRA::Live version.

These layouts consists of a multi-channel “FFT” analyzer. It aims to analyze an **audio system**, more specifically a **linear and time-invariant** [1] one. One channel serves as a reference, while the others are compared to it.

In live system tuning, we usually excite the sound system with a known signal, such as pink noise or a logarithmic sweep. The reference is the signal itself, while the other channels receive the input from different microphones placed in the venue we want to measure. The multi-channel FFT then provides us with the **transfer functions** and the **impulse responses** of each channel.

The transfer function provides two key pieces of information:

- The **magnitude curve**, which describes how the system amplifies or attenuates frequencies.
- The **phase response curve**, which describes how the system delays frequencies.

The impulse response describes how the system reacts to an ideal impulse. An ideal impulse is also known as a Dirac impulse. This representation allows us to visualize the time of arrival of the different signals.

The following sections will provide detailed information on how to set up for the best possible results.

The behavior of the session system will be explained in order to help you organize your work, and how the IO relates to this specific layout.

Then, we will discuss the two main scopes used in the layout: the transfer function and the impulse response.

Of course, we will also explain the structure of the user interface and how to work efficiently inside this system tuning page.

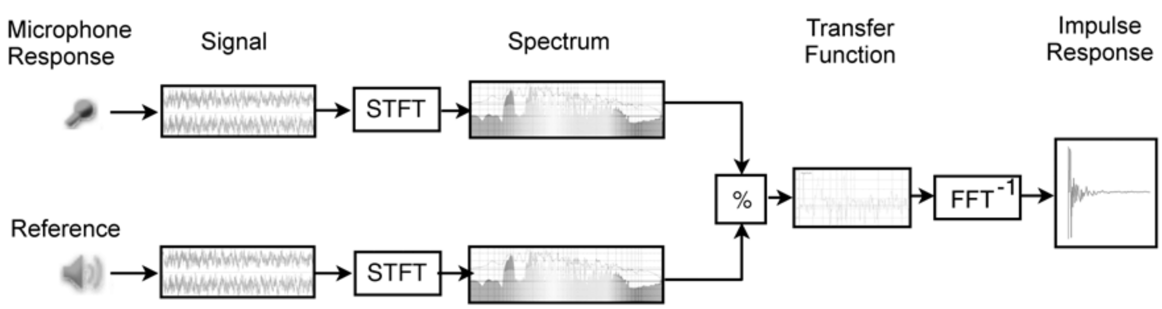
[1] A linear and time-invariant system must meet two criteria: the processing of signals x and y must be the same as the sum of their individual processing (linearity), and the processing of a signal at any given time must always produce the same output (time invariance). Rooms, reverb units and speakers generally function linearly. However, compressors, saturators and limiters do not. Chorus, tremolo, vibrato, and modulation effects are not time-invariant. See more.

27 Initial Setup

Throughout this documentation, we will refer to the measured signal processing chain as the system (sometimes called device under test in electronics literature). This system input is fed with a source signal, which produces a response signal at its output(s). Both source and response are recorded and monitored by the analyzer, from which several measurement curves are produced.

The first step is, therefore, to set up the measurement chain. In cases where an outboard or plugin device's characteristics are to be measured, this is just a matter of routing the inputs and outputs in your DAW.

If you're measuring the acoustic response of a physical space, you'll need to place at least one microphone at the preferred listening position to record the response. The source can either be picked up directly at the DAW output or recorded with a second microphone placed in front of the loudspeaker(s), depending on whether you want to include the loudspeaker's influence or not in the measurement.



> **Note**
>
> System analysis overall principle.

27.1 Practical considerations for capturing measurement signals

At first glance, an audio signal chain is very much like a series of black boxes. As an audio engineer, you can trust your ears and the manufacturer's data sheets to assess the effects this chain has on the incoming audio. In a variety of cases, however, this is either simply impractical, not possible or not precise enough. Such situations include live sound setups, recording setups, etc., where unknown factors such as the venue's or studio's acoustic response are a crucial part of the chain.

It is therefore necessary to resort to scientific measurement procedures and tools to obtain precise, trustworthy and reproducible results. The main tools at your disposal for this purpose are transferring curves and impulse response measurements, which are especially designed for this task.

As with any measurement instrument, it is important to have a good grasp of its mode of operation as well as any possible limitations in order to use it most efficiently. Some knowledge of acoustic principles and notions of signal processing are naturally required as well. While this manual tries to cover most typical use cases and point out common dos and don'ts, it obviously cannot replace either a good textbook or practical experience.

27.1.1 Use a measurement microphone

The goal here is to take the measurement chain out of the equation, so only specially designed microphones that exhibit a flat curve, minimal coloration, lowest noise and distortion should be used.

27.1.2 Choose a neutral preamplifier and calibrate it accurately

For the same reasons, select the most neutral preamplifier and A/D D/A convertors you have at your disposal. It is especially important to be able to set accurate and reproducible gain, linear and flat responses. Take special care that the signal is not so hot as to clip or distort the preamplifier input stages, as this would distort the measurements accordingly and induce you into error.

27.1.3 Maximize signal-to-noise ratio

When measuring an acoustic system, raise up the volume as high as practical for maximal signal-to-noise ratio, and try to minimize any spurious acoustic noises such as footsteps and conversation. As always, the goal is to set the test signal as high as possible above the noise floor while ensuring all devices still operate in their linear region. Finally, make sure the microphone is firmly held in position and acoustically decoupled from the floor.

In a live concert context, especially with the audience present, using a noise signal is not practical. In this case, you can still perform measurements using a music signal, but the measurements will be less accurate as the signal isn't known in advance and does not necessarily contain all frequencies like noise does.

27.2 Measurement setup

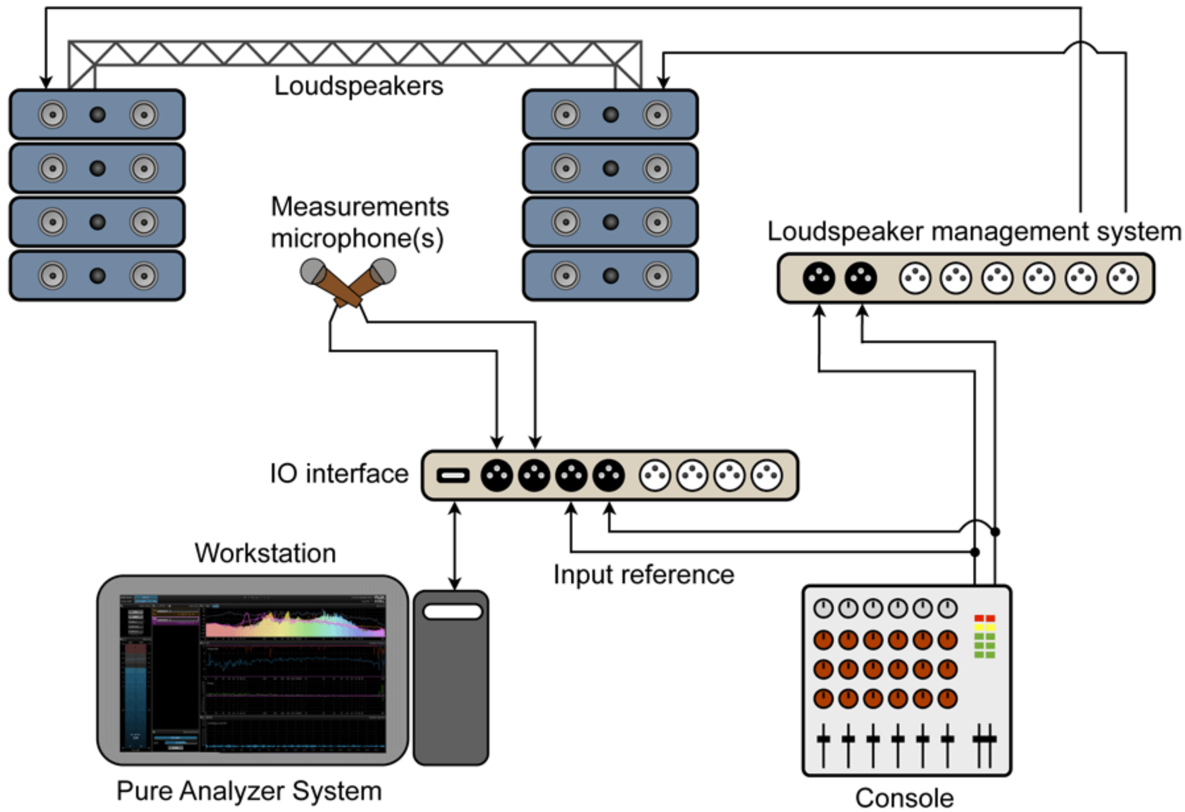


Figure 27.1: Typical configuration for a live venue measurement setup using an external signal generator.

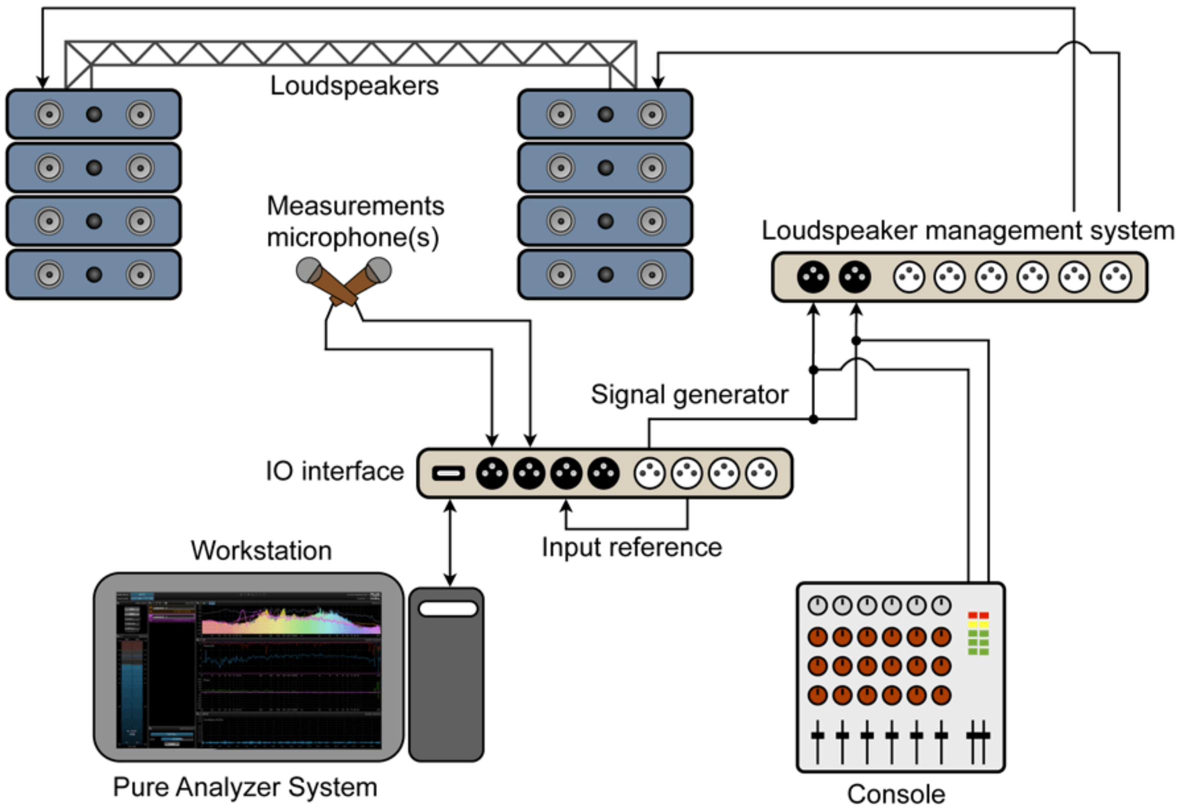


Figure 27.2: Typical configuration for a live venue measurement setup using MiRA's internal signal generator and loopback

27.3 Test signals

FLUX:: MiRA:: is designed to cover the broadest range of practical use cases, and does not impose a limitation on the measurement signal used.

Traditionally, transfer curve and impulse response measurements are performed by feeding a specially designed test signal into the system, the most commonly employed being pink and white noise and swept sines. While these types of signals are those that give the best and most accurate results, with each having its own strengths and weaknesses, they do prohibit the measurement of a system in the context of a live system with the audience present.

Performing measurements using a live music signal allows the engineer to fine-tune the system settings to compensate for changing conditions, such as the effect of the crowd on acoustic reflections and damping, varying temperature and humidity, etc.

Important

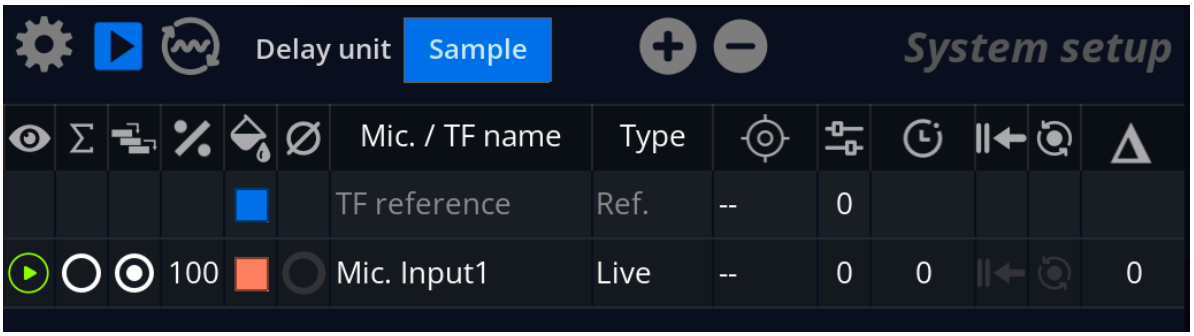
Although less pleasing to the ear, we do, however, recommend using a noise test signal whenever possible, at least as a starting point.

You are free to use any kind of test signal generator, outboard or plugin, provided you trust it being reliable and easy to use. A selection of plugins suitable for this task is shown in the chart below.

Note

While FLUX:: MiRA:: does not impose any limitation on the test signal used, we recommend using the integrated Signal generator 31, which has been especially designed for this task. We conducted thorough tests on a wide panel of signal generators available as plugins or integrated into DAW software and found that many do not meet the requirements for performing accurate and reliable measurements.

28 System Setup



The system setup UI scope lists all the inputs from the live system, along with their associated options. Here, you will be able to name the input, compensate for delay offsets, apply a target reference, associate a floor and a head microphone, etc. Each input gives four curves:

- The spectrum magnitude
- The phase response
- The coherence curve
- The impulse response.

28.1 Basic operations

The visibility of an input is defined by the toggle in the column.

To rename an input, double-click on its name in the 'Mic./TF name' column.

To change the color, double-click on the colored square in the column.

28.2 Gain & target

The column allows for adjustment of the gain of each capture, in decibels.

The button toggles phase inversion of the selected channel.

A reference target for the input can be defined in the column.A target is a previously recorded capture that is used as a reference for the input. For example, it is often used to calibrate measurement microphones.

28.3 Computation Curve

The computation curve appears automatically in the system setup list if more than one microphone is present. To feed an input into the computation curve, you must check its $\sum$ column.

The computation curve has three different computing algorithms, which can be accessed in the "type" column.

- The *averaging* mode is recommended when using several microphones at **different locations in the same venue**. When using this algorithm, you can adjust the weight of a microphone by adjusting the % column. One hundred percent indicates full contribution, while zero percent indicates no contribution. The coherence score also affects the computation curve: a measure with low coherence will have its weight lowered. You can deactivate this behavior by unchecking the combo box in the column.
- The *sum* mode simply adds the magnitudes of the different curves.
- The *acoustic mode* summarizes the magnitude, but it also takes phase relationships into account. This mode is recommended when dealing with separated measurements for **heads and sub speakers**.

28.4 Delay finder

FLUX:: MiRA:: uses an automatic delay-finding algorithm to determine the time-of-arrival difference between several microphones and the reference input.

The delay unit can be chosen from the drop-down menu. Possible options are:

- Delay in samples (smp).
- Distance in meters (m) or imperial feet (ft.).
- Delay in milliseconds (ms).

The delay finder **always tracks** for delay changes. The detected offset is displayed in the Δ column in red.

28.4.1 Delay compensation

Pressing the button activates a delay line in the source signal path, compensating for the currently displayed delay value. This effectively aligns the *source* and *response* signals.

If necessary, you can manually adjust the delay figure using either of these methods:

- Direct keyboard numeric value entry as time or distance figure.
- Increment / decrement by clicking the +/- icons.
- Increment / decrement using the +/- numeric keys.

28.4.2 Delay adjustment considerations

Ensure stable conditions while performing a measurement

You should ensure both source and response signals have reached stability before attempting measurement. In particular, do not stop or start the audio, change the volume or any other parameter just before or during measurement. This would invalidate the measurement and you would have to start again.

Limitations

Please note there are many unknowns in play when determining the optimum delay figure. While we did our best to make this tool as robust and accurate as possible, there is always a possibility that it will fail, as with all automatic procedures. In this case, you should repeat the process or resort to manual adjustment until you get satisfactory results.

Multiple Paths

The major assumption behind delay compensation is that there is a main direct path from source to listener. This obviously does not apply in a very reverberant or complex-shaped acoustic space. This is where acoustic expertise and trial and error come into play in order to attain the best compromise.

28.5 Input Type

An input microphone can be of three different types. By default it is considered as a “Live (full band)” input.

The other accessible type is “Live Floor Mic (pair with..)”. When a microphone is switched to this type, it is expected to be placed on the floor to reduce the influence of floor reflection in the measurement process. It is then paired with another microphone. The one on the ground will produce data for the low frequency content of the measure, while the paired microphone will produce the data for the high frequency content. The crossover between the two microphones can be set in the configuration menus of the system setup.

The reference input displays a type of "Ref." and cannot be edited.

The computation curve uses the type to define its averaging algorithm. See the section above.

28.6 Header buttons

For common scope header buttons, see the Audio Analysis Scopes section.

When the is engaged, it loops the first output channel of the signal generator into the reference input of the system setup. This setting is also accessible from the IO configuration menu.

The "Delay unit" drop-down allows for changing the delay unit, as seen in the delay finder section above. The buttons allows for changing the delay value of the select input of ± 1 sample.

28.7 Options menu

The options menu is accessible by clicking on the button.

The **Delay Finder FIFO** parameter sets both the maximum findable delay time as well the global time averaging of the delay finder. It should be left to its default setting.

History time defines the length of the analysis buffer. It is set to 5 seconds by default.

Pairing crossover freq. sets the crossover point for floor/head microphone pairs.

Auto-pause: when the signal is below the level indicated on the Main IO, the Transfer Function will not be processed.

Columns visibility allows you to show/hide specific table settings.

29 Sessions & Captures System

29.1 Introduction

MiRA:: has a very complete system for managing all the measurements done by the user. Inside the application, a measure is called a **capture**. A capture contains the channel's **spectrum**, **transfer function** and **impulse response**. Captures are organized per **session**. Each session is materialized by a `.fcap` file inside the following folders:

- `~/Library/Application Support/FLUX/MiRA/Captures` on macOS
- `C:\User\/...\AppData\Local\FLUX\Captures` on Windows.

A session can serve to multiple purposes. It can represent the measures of the sound system of a venue or of a specific audio equipment. It can also serve to store calibration files or target curves.

> **Note**
>
> MiRA:: can import a wide variety of calibration and measure files from other manufacturers.

For system tuning, the main workflow strategy of MiRA:: is to allow very **quick system measurement** while giving you all the tools to **work and recompute offline**.

To accomplish the on-site measurement, you can use pink noise to calibrate the delay finder algorithm, then use a few sweeps to measure what you need to measure (heads, subs, front-fields, etc.). Note that MiRA:: has its own signal generator to simplify the process.

Once the captures are acquired, **MiRA:: can recompute delays, average, acoustic sums and so on without having to keep playing sounds through the sound system**

29.2 Quick start to captures

29.2.1 I/O setup

The first step of any capture process is to make sure that the I/O is correctly configured. Go to the IO menu, by reaching either the info header, or to the top menu `Mira>IO settings` on macOS, or `Files>IO settings` on Windows.

Make sure to select the right input and output audio interface. Then, you must configure the different input channels of the capture system. In the case of this example, the **number of microphones is set to one**, then the first line of the table is set to **input channel one** and as the **transfer function reference**. The second line of the table is set to **input channel two** and as a **microphone input**.

With these settings, our reference signal will be detected on input channel one and our measurement will come in input channel two.

We also need to make sure that the signal generator is correctly patched. In our example, we want the signal generator to output on channels one and two. **Same signal is outputted on both channels**, it simply makes the next part of the routing easier. The output 1 of our audio interface is looped back to input 1. The output 2 goes to our system, and the output of the system goes to input 2 of our audio interface.

At this point, our whole routing is set up.

29.2.2 Creating a session

The very first step is to create a new session to store our captures. Simply click on the "new session" button. A popup appears to ask you to name the session. Name it accordingly and press OK.

29.2.3 Checking for delay offset

It is **absolutely crucial** to make sure that all your microphone inputs are properly aligned with the reference signal. In MiRA::, the recommended method is to set the signal generator to a pink noise and start it. If our previous routing is good, we should now see some time offset appearing in the δ column of the system.

You can apply these delays by clicking on the button.

29.2.4 Acquiring the capture

It is of first importance to understand that there are two main ways to create captures:

- You can send **any signal** into the system to measure, then create a capture when the curves have stabilized.
- You can **automate** the capture to the emition of a sweep tone.

Most of the time, it is preferred to use an automated sweep tone to realize the measure. In MiRA:: we simply have to click on the icon. Each time, MiRA:: will ask you to name the capture.

Repeat the process for each measure that you need to create (different loudspeakers, heads, subs, front field, etc.).

29.3 Capturing online, processing offline

In the previous section, we have seen many parameters related to the different input channels. It is very important to understand that none of them are permanently bound to the measure itself. You can always modify the applied delay, capture & target curves, microphone pairing and gain.

The computation curve itself is automatically regenerated based on the modification made to the captures. The main idea is, as stated above, to captures online and to process offline. In other words, most of the system tuning job can be done without being tied to the PA system itself.

29.3.1 Offline settings

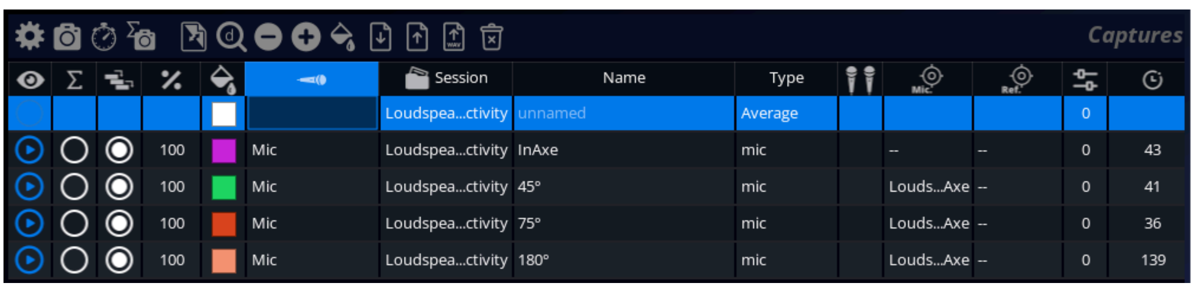


Figure 29.1: Captures menu

All the offline adjustments and settings are done in the capture menu. Because they are identical to the System Setup menu settings, please refer to this part of the user documentation.

29.3.2 Computation curve and sessions

A session can have as many captures as the user want, but only one computation curve. It is because the computation curve is dynamic, depending on which captures are affected to it, the time alignment between them, and so on.

Still, it is possible to store the current state of the computation curve to a new capture, to keep track of your improvement, to access quickly to different computations or to store them as target curves. To capture the computation curve, click on the *Capture Computation Curve* button.

29.3.3 Changing capture settings

By default, newly created captures follow the settings of the transfer function. If you want to change the spectrum type, block size or time averaging afterward, you can use the "Update capture setting" option in the "Capture" menu in the top menu of the application.

30 Microphone Pairing

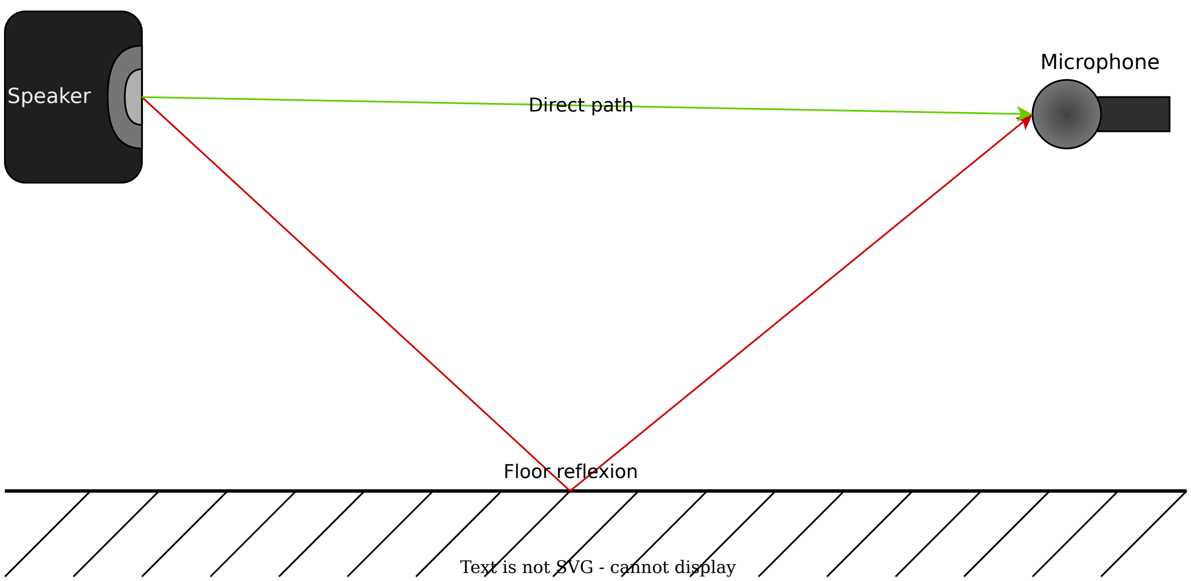
30.1 Overview

MiRA's microphone pairing feature is an innovative method for optimizing audio system tuning by addressing low-frequency floor reflections that can distort measurements. It simplifies the measurement process by intelligently pairing microphones and combining their data in real time. This approach enhances accuracy and eliminates the need for extensive post-processing or additional DSP (Digital Signal Processing) hardware. As with all other MiRA system analysis features, pairing options can be revised offline after capturing all measures.

30.2 Technical Description

30.2.1 Problem Statement

One common challenge when tuning a Public Address (PA) audio system is dealing with low-frequency reflections off the floor. These reflections can interfere with the direct sound, causing comb-filtering and leading to inaccurate measurements.



Traditional solutions involve conducting separate measurements with microphones placed on the floor and at head height, and then manually analyzing the acquired data separately. An alternative method employs two microphones equipped with FIR (Finite Impulse Response) filters (low-pass and high-pass) that are summed, necessitating additional expertise and equipment, and introducing latency. These techniques are laborious, time-consuming, and technically intricate.

30.2.2 MiRA:: Microphone Pairing Feature

The proposal solution uses a pair of microphones, one on the floor and one at head level. Both microphones are configured in MiRA software, which automatically combines data from the paired microphones. This eliminates the need for multiple captures and manual data merging, streamlining the workflow. Additionally, the software supports multiple microphone pairs for simultaneous multi-point measurements.

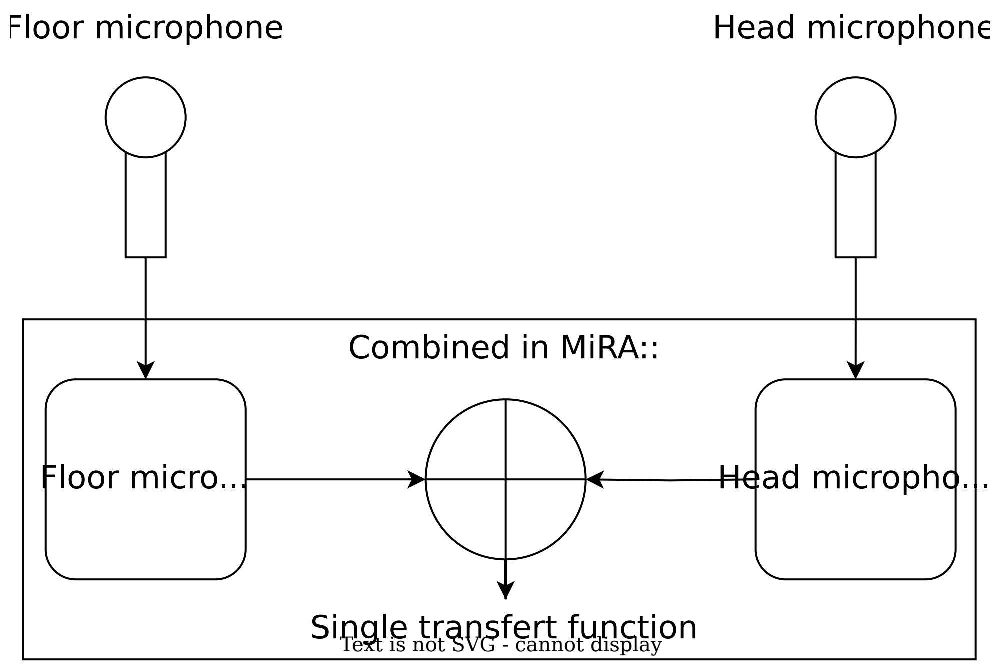
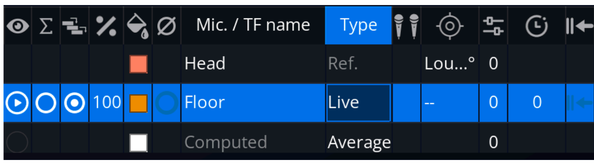


Figure 30.1: Installation Schematic

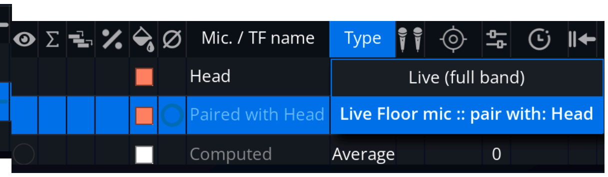
Despite being real-time measurement software, MiRA allows users to edit their measurements and captures. Users can modify alignment delays, microphone gains, calibrations, and other parameters even after the initial recording.

30.2.3 Workflow

To begin, connect the microphones and position one near the ground, whilst keeping another at hear level. In MiRA, each microphone is recognized as “Live (full band)” by default. Change the floor microphone type to “Live floor mic” and pair it with the head-level microphone.



(a) Click on “Live” to open the pairing drop-down menu



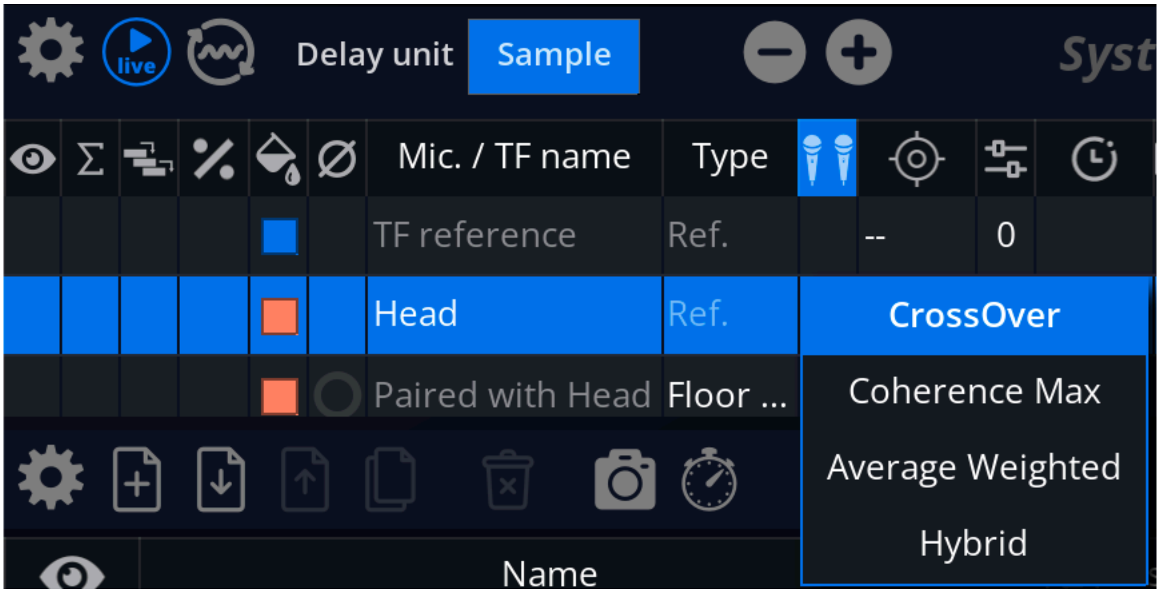
(a) Selected the desired head microphone

MiRA captures data from both microphones simultaneously and automatically combines data from both microphones. Users can edit measurements after capture, adjust alignment delays, microphone gains, calibrations, and other parameters within MiRA. The result is a single transfer function curve for each microphone pair, with the impulse response corresponding to the head-level microphone.

> Tip
>
> For small venues or quick setups, using one pair of microphones, as described, simplifies the measurement process without additional complexity. For large venues requiring comprehensive coverage, deploying **multiple microphone pairs at different locations** allows simultaneous multipoint measurements, saving time.

30.3 Pairing algorithm

MiRA:: offers four different modes for microphone pairing.



The first one, called **CrossOver**, is a frequency-based approach. The floor microphone is used for frequencies below the crossover frequency, while the head microphone is used for frequencies above. The crossover frequency is user definable.

The **Coherence Max algorithm** selected the microphone with the greater coherence for each analysis band.

The **Average Weighted** algorithm combines the signals from the two microphones based on their coherence scores for each analysis band. The higher the coherence score, the more weight is given to that signal in the computations.

The final mode is '**Hybrid**', which combines an **Average Weighted** magnitude curve and a **Coherence Max** computation for the phase and coherence curves.

31 Signal generator

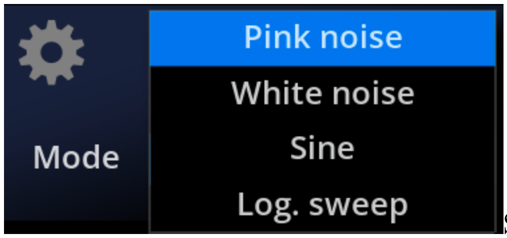
The signal generator "scope" produces commonly used signals for system measurement.

31.1 Routing

The signal has two output channels. Usually, one loops back directly into the audio interface and serves as the reference. The other one goes to the system we want to measure. Note that the exact same signal is sent to both outputs. The output channels settings are in the IO setup menu.

> **❗ Important**
>
> It is also possible to use a "software loopback" instead of a "hardware loopback" by clicking on the button in the system setup scope.

31.2 Controls

Name	Description
Type	 Sets the signal type to generate.**Pink noise**Pink noise is a random signal with an amplitude falloff inversely proportional to frequency. This is the most commonly employed variety noise in audio measurement, as it is a constant-energy perceived content.**White noise**White noise is a random signal with constant energy across the audio range. Compared to pink noise, it sounds much brighter as it has more energy in high-frequencies. Commonly employed for electronic apparatus measurements.**Sine**Fixed-frequency, pure tone generator.**Sweep**Generates a variable tone from start to end frequencies. It is a Log. sweep, as it is best suited for audio measurements due to its constant time per octave.
Level	Output level of the waveform, expressed in dB RMS.
Enable	Toggles signal generator output on and off.

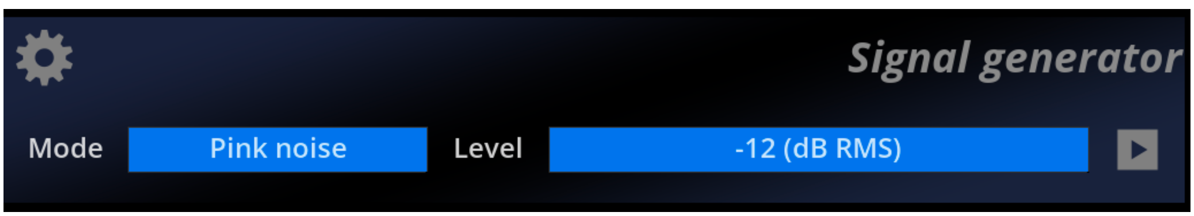


Figure 31.1: Signal generator controls.

31.3 Settings

31.3.1 Feed input reference

Name	Description
Feed input reference	Fed the reference input (default input 1) with the signal generator.

Name	Description
Sine frequency	Sets the frequency of the sine generator, only applicable when the signal type is set to Sine.
Sweep start/end frequencies	Sets the range of frequencies to sweep. [1]
Sweep length	Sets the overall duration of the sweep in seconds, i.e. the time taken to go from start to end frequency.
Level	Generator output level in dB RMS.

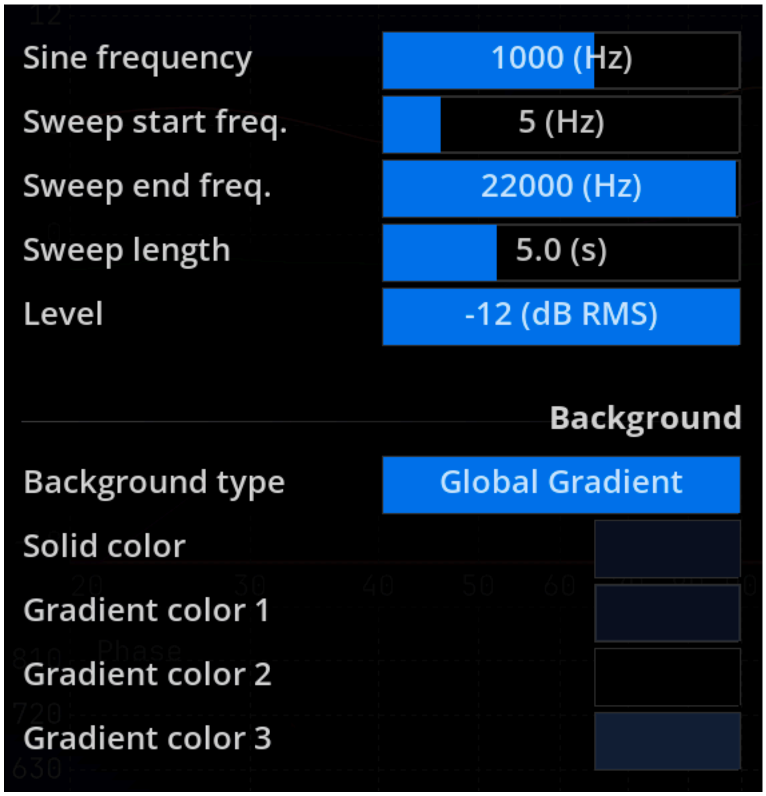


Figure 31.2: Signal generator setup options.

[1]Reverse start and end frequencies to obtain reverse sweep.

32 Transfer function

32.1 Introduction

The transfer function of a system measures its frequency response, which is expressed in terms of magnitude and phase response. The transfer function measures how the system affects the magnitude and phase of an incoming signal at different frequencies, and is essentially a ratio of output versus input spectra.

Practical uses of this are numerous: determining the curve of an equalizer, determining what frequencies are emphasized by an outboard device, measuring a room's acoustic response, etc.

> **Note**
>
> The transfer function assumes the system is linear & time-invariant. Linearity notably implies the system is free of distortion, and time invariance that its characteristics do not change in time. Failing to meet these requirements will lead to unpredictable results.
> In practice, the transfer function is considered an adequate measurement technique for most real-world systems, except for devices exhibiting highly non-linear behavior, such as compressors and distortion effects, and time-modulation based effects, such as chorus and flanger.

32.2 Transfer function magnitude

The transfer function magnitude displays the gain versus frequency curve of the system under test. A passthrough obviously results in a flat horizontal line centered on 0dB. This line represents the ideal curve one would be able to achieve if all the system defects could be compensated for, and that serves as a reference target when doing room correction.

32.3 Transfer function coherence

The coherence is a normalized - that is comprised between zero and one - measure of the confidence of the transfer function at a specific frequency. In other words, it describes how trustworthy the transfer function is at the corresponding frequency.

Coherence at a particular frequency indicates whether the system can accurately be described as linear gain and phase shift or not.

32.3.1 Interpretation and uses

A low coherence most often indicates a bad measurement, so you should look for possible causes and correct them before starting again. Improper delay compensation leads to low coherence results, so this is the first thing to check. Other typical culprits include a noisy device, the presence of distortion, channel crosstalk, acoustical noise such as cooling fans, people talking, handling noise, bad isolation from the outside, etc.

While maximizing the coherence is desirable, in most cases, it will most likely be impossible to attain a flat curve approaching unity at all frequencies, except in an anechoic chamber or very 'dead' sounding room with minimal reflections.

Reverberation, as well as mismatched transducers, tend to give lower coherence, as the signal arriving at the microphone position is really the sum of several time-delayed versions of the source.

Sometimes, it will be impossible to get good overall coherence, and the magnitude and phase curves will, therefore, be less precise, stable and smooth. This does not mean you cannot attempt to extract any information from those. As always, use your judgment and knowledge of the specific system to decide which assumptions seem reasonable.

32.4 Transfer function phase

Phase information is sometimes overlooked, and indeed it is less straightforward to understand and interpret than magnitude. Altering the phase of a signal can range from subtle to dramatic, and phase distortion can lead to temporal smearing of the audio, loss of spatial information, and other nuisances.

The transfer function phase curve displays the phase difference between the system's output and input at different frequencies, in degrees, ranging from -180 to 180.

> **Note**
>
> FLUX:: MiRA:: employs several smoothing algorithms custom designed for phase curve smoothing, as explained in the section about phase setup.

Due to the definition of phase itself and the means of computing it, the curve is generally more sensitive to extraneous noise, distortion and time-varying conditions.

Even more so than with the magnitude curve, a precisely compensated delay is critical to accurate phase computation. In very reverberant environments, the phase curve will be very chaotic. This is inevitable and a direct consequence of the complex nature of the system, and not a limitation of the instrument.

> **!** Important
>
> We advise using ART analysis mode, which mitigates phase computation inaccuracies compared to plain FFT.

Figure 32.1: Typical transfer function display in a live room!

32.5 Settings

Time averaging is on by default, as the goal here is to provide the most stable display, and to eliminate any variations of the signal in time.

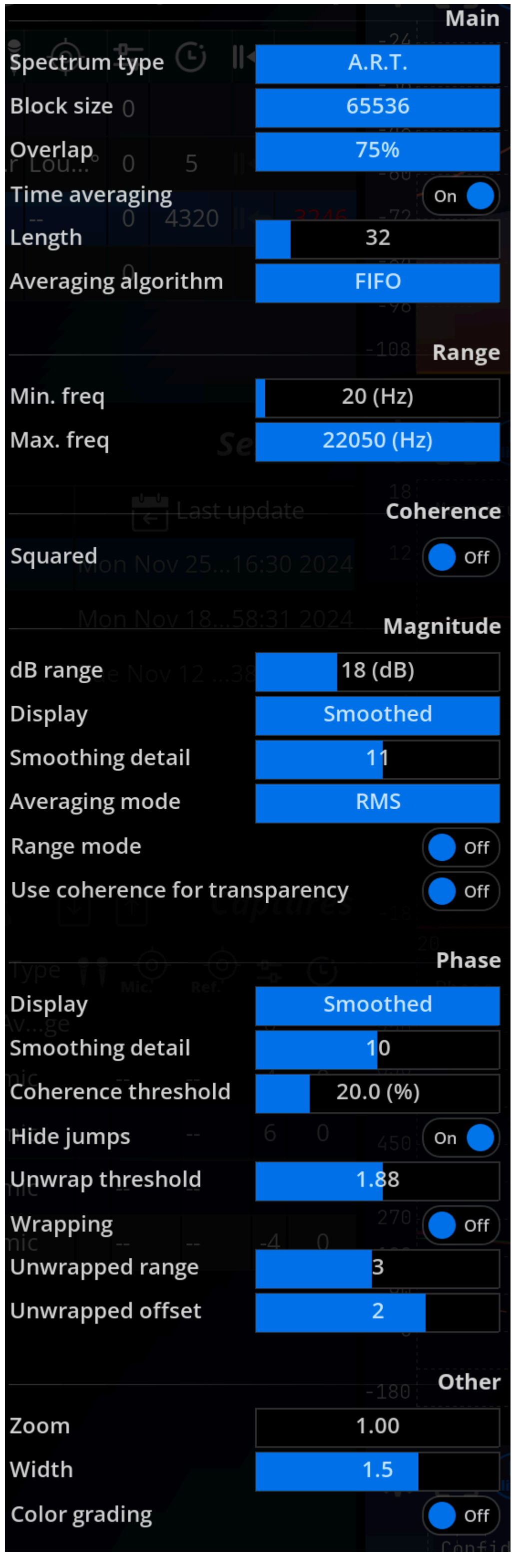
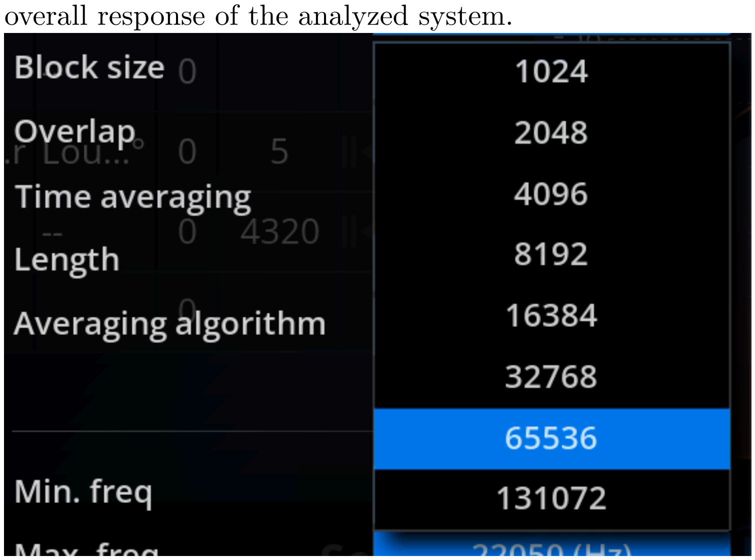


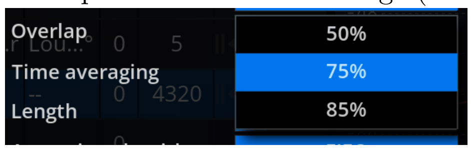
Figure 32.2: Transfer function setup options

Frequency smoothing can be useful to smooth out irregularities and get a general picture of the curve. It is advised to use this function sparingly, though, as it can change values by a large proportion, and obscure potential problems with either the actual system being measured, or the measurement setup itself.

A combination of time averaging and frequency smoothing is most often required to obtain readable results in real-world scenarios, especially in large rooms and outdoors.

32.5.1 Main

Name	Description
Spectrum type	Three options are available: - FFT, which is the classic Fast Fourier Transform. The Fourier Transform is linear in frequency, so it tends to give too much detail at the higher end of the audio spectrum. - Variable Q, which provides a measure more coherent toward our perception of frequency - ART, which runs multiple FFT with different window sizes to get a good time resolution and a good frequency resolution. The default, and recommended option, is ART.
Block size	Block size used for the transfer function and the capture done with sweep. The default is 65536, which is appropriate for most cases. Increasing this value gives better frequency resolution, at the expense of CPU load. Lower values can be employed if you're only interested in the overall response of the analyzed system. 

Name	Description
Overlap	The overlap mode setting determines how much incoming audio frames overlap each-other. A higher overlap results in a smoother display update, at the expense of increased CPU usage. The available settings are: 85%: highest overlap 75%: medium overlap size. 50%: minimized overlap for minimal CPU usage (useful for slow machines) 
Time averaging	Toggles time averaging on and off. Default is on, which, in most cases, is necessary to provide a stable display readout.
Length	This setting determines the number of blocks taken into account to compute the averaged transfer function. Increasing this value will give a smoother readout, but the display will react more slowly to any input variations, and CPU load will be higher. The default is 32.
Averaging algorithm	The averaging can follow two different algorithms: - The FIFO mode (first-in first-out) corresponds to a simple sliding averaging. - The exponential mode is equivalent to a first order low-pass filter.

32.5.2 Range

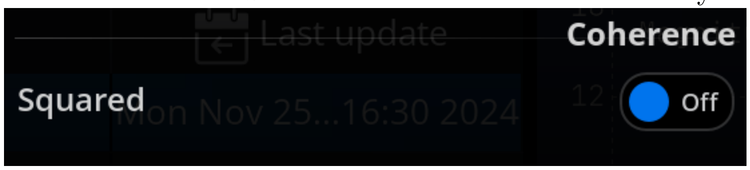
Name	Description
Min. Freq	Minimum frequency display by the transfert function
Max. Freq	Maximum frequency display by the transfert function

32.5.3 Coherence/magnitude

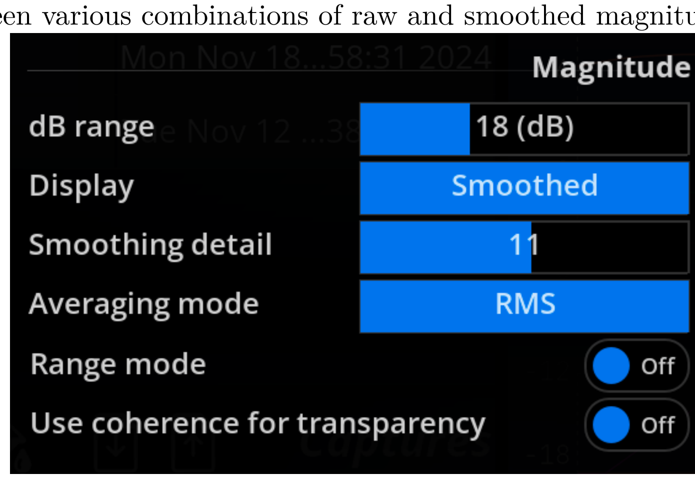
Name	Description
Smoothing detail	Sets the amount of detail present on the smoothed magnitude and coherence curves. This number is roughly the maximum number of valleys and peaks that will remain after smoothing. A low value of around 10 is good for getting a global and uncluttered picture of a room's frequency response. [1]

[1] Relying on smoothed curves altogether should be avoided, as smoothing can mask-out essential information such as room modes, which materialize as sharp peaks and dips in the transfer function magnitude curve. We strongly recommend basing your judgment on both raw and smoothed curves even when the raw curve is very noisy.

32.5.4 Coherence

Name	Description
Squared	Apply a square function to the coherence values.
Display	Toggles between one of three modes: - Full : main unsmoothed coherence curve. - Smoothed: smoothed coherence only. - All: both. 
Color	Color of the pen used to draw the coherence curve.

32.5.5 Magnitude

Name	Description
Range	Minimum and maximum values to which the display is clamped, in decibels.
Display	Toggles between various combinations of raw and smoothed magnitude curve display.  - Full : main unsmoothed magnitude curve. - Smoothed: smoothed magnitude only. - All: both. Keep in mind the smoothing process can filter out a lot of information, so relying solely on the smoothed curve should be avoided.

Name	Description
Smoothing detail	Adjust the amount of detail displayed in the curve. Lower values imply more smoothing. This setting should be set so that the curve shows meaningful and accurate information. Too little details will induce a loss of information, while too many details will display information that may be inaccurate.
Averaging mode	Choose the averaging mode of the transfer function magnitude. Vectorial mode computes the average sum of magnitudes and magnitudes multiplied by coherence. In vectorial mode, the averaged magnitude is therefore an indication of the perceived magnitude spectrum, i.e. the sum of the direct path and diffuse field signals. RMS mode computes the average as the root of the sum of the square magnitudes. Default is RMS.
Range mode	Toggles auto-range on and off. When enabled, the display range automatically follows that of the transfer function magnitude curves, which is useful for hands-free operation, for example. Default is off.
Use coherence for transparency	Allows to use the coherence values to define Magnitude.

32.5.6 Phase

Name	Description
Display	Toggles between the various phase curve display modes: - Full: raw phase only. - Smoothed: smoothed phase only. - All: both.
Smoothing detail	Adjust the amount of detail displayed in the curve. The widest fractional octave values give a smoother curve. This setting should be set so that the curve shows meaningful and accurate information. Too little details will induce a loss of information, while too many details will display information that may be inaccurate.
Coherence threshold	Mask phase curve region where the coherence is below this threshold
Unwrap threshold	Raise to unwrap larger phase jumps only (only relevant in tolerance mode)
Hide jumps	When enabled, the portion of the curve that corresponds to a phase rotation is not displayed.
Wrapping	Display the phase either wrapped between -180° and 180° or unwrapped.
Unwrapped range	Adjust the phase range, for $-n \times 180$ to $n \times 180$, where n is the value of this settings.
Unwrapped offset	Increment an offset of $n \times 180°$, where n is the value of this settings.

32.5.7 Other

Name	Description
Color grading	Apply frequency-dependent coloring to the curve. Default is off.
Zoom	Curve zoom ratio slider.
Width	Size of the pen used to draw the coherence curve.

33 Impulse response measurement

33.1 Introduction

The impulse response of a system is the signal obtained at the output when feeding a click (also termed impulse, spike or Dirac) its input. It is a fundamental tool to describe the time properties of a linear system.

Combined with the transfer function, impulse response measurement is essential in characterizing the acoustics of a studio, concert hall or venue, from which synthetic figures such as reverberation time are derived. Determining the impulse response of an amplifier and loudspeaker in tandem can also serve to assess their performance.

A pass-trough device, or equivalently, a completely dead space such as an anechoic chamber, exhibits a unit impulse response, whose value at zero time is gain, and is zero at all other instants.

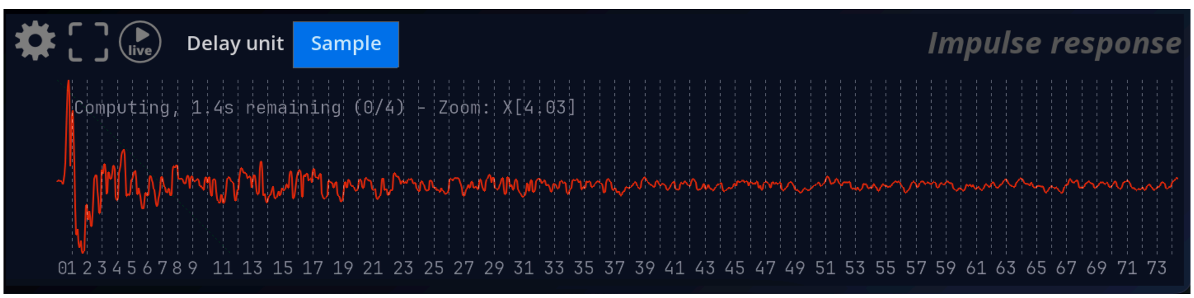


Figure 33.1: Impulse response display example

33.1.1 Analyze/freeze

The  button toggles the impulse response real-time update on and off.

33.1.2 Delay Unit

Change the unit in which the time delay is expressed. It can be expressed as:

- Samples
- milliseconds
- meters
- and feets

33.2 General procedure

Impulse response (IR) measurement requires that sufficient samples be accumulated before the actual computation is ready, depending on the values of the Max Length and Time averaging 33.3 settings. The user interface displays a message indicating the remaining time before the display is ready, whenever the related settings are changed or the reset button is pressed.

Because the software cannot detect whenever you make changes to the analyzed system, you need to press the Reset button in the setup or wait for the display to stabilize before reading the display.

Once your test setup is ready, press the 'Reset' button and wait for the display showing the remaining time to disappear, at which point the IR display is ready. When a sufficient amount of samples has been accumulated, IR computation goes on as long as the 'Run' button is active, and updated with new incoming samples.

> **Note**
>
> Make sure the actual impulse response is shorter than the maximum specified time, otherwise mild to severe time-aliasing will occur, and the measurement will not be reliable.
> A good rule of thumb is to set the Max length parameter to twice that of the estimated RT60 of the room.
> If in doubt, raise the Max length setting until the impulse response curve does not change, and check the tail of the curve does indeed fall to zero.

33.3 Time averaging

The time averaging function computes the mean of several IR measurements over time, which is very useful to filter out noise and other artifacts. It is enabled by default as this gives better display stability and measurement robustness, however averaging also slows down the reactivity of the display to incoming variations, so you can disable it if needed.

When IR averaging is enabled, a message is shown giving the number of currently computed impulse responses versus averaging length. The display switches to show the mean confidence percentage when ready.

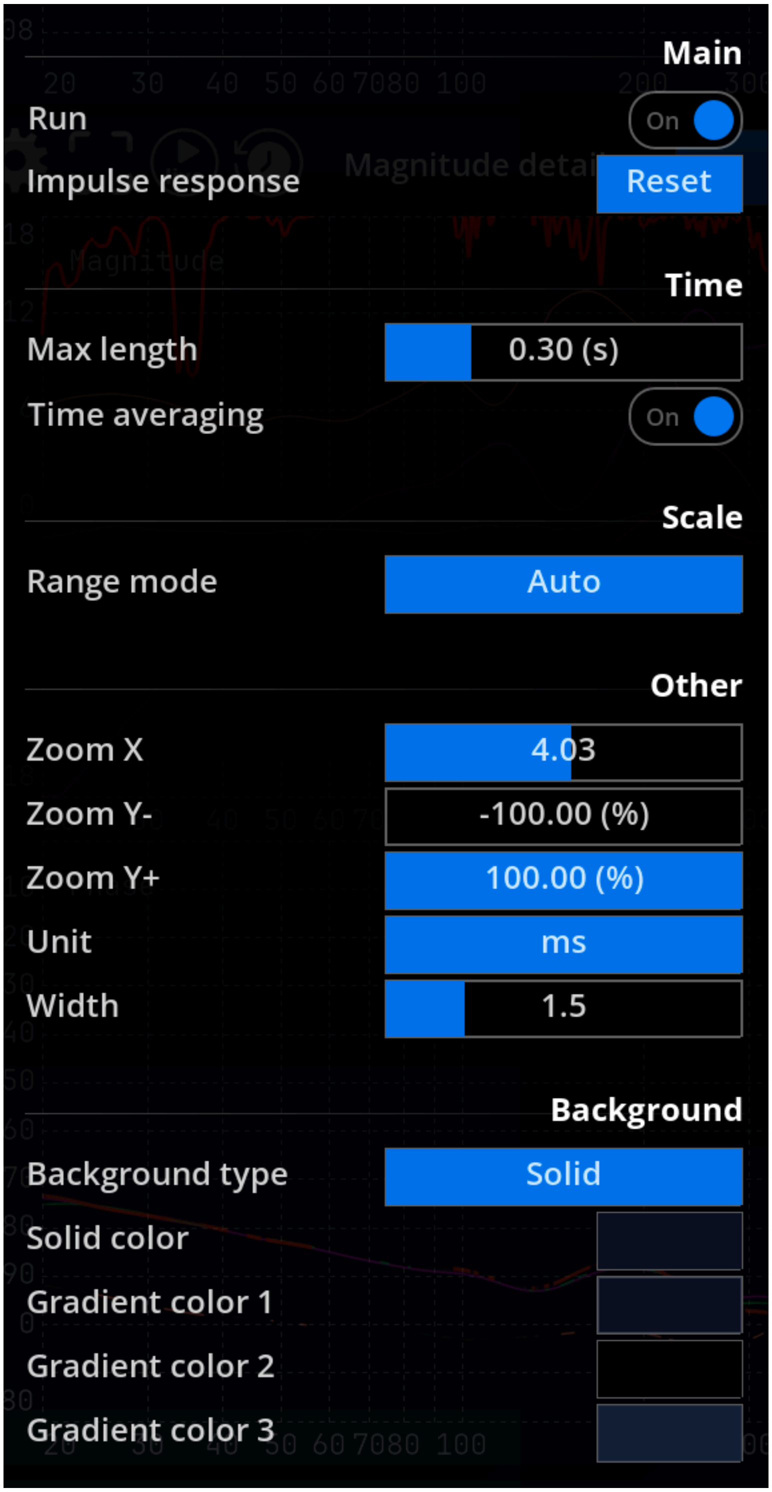
33.4 Settings

33.4.1 Main

Impulse response setup options

Name	Description
Run	Toggles impulse response live update on and off. Default is on. You can temporarily freeze the impulse response with this button, to examine it in detail at your leisure, without worrying about changing external conditions. Disabling 'Run' is equivalent to freezing the measurement, and leaves the averaging buffer as is.
Reset	Resets the impulse response computation, including the averaging buffer.[1]

[1] If you are using a lengthy averaging setting and have just changed your setup, you can reset the entire impulse response to immediately forget previous measurements.


Main
Run
On
Impulse response
Reset
Time
Max length
0.30 (s)
Time averaging
On
Scale
Range mode
Auto
Other
Zoom X
4.03
Zoom Y-
-100.00 (%)
Zoom Y+
100.00 (%)
Unit
ms
Width
1.5
Background
Background type
Solid
Solid color
Gradient color 1
Gradient color 2
Gradient color 3

33.4.2 Time

Name	Description
Max length	Sets the maximum length of the impulse response in seconds. If the reverberation time in your room exceeds this value, time-aliasing will occur, meaning that the impulse response computation will be incorrect and some of the reverberation tail might end up at the start of the display. The default value is 0.3s. Increasing this value not only requires more processing power, it also increases the time needed to wait for the display to be updated, as the calculations involved need a greater amount of incoming audio samples to be processed. Combining time averaging and a long length setting means you'll have to wait 30 seconds or so for the display to stabilize, so you should really do this if you need to or do not mind waiting.
Time averaging	Accumulates several impulse response measurements and averages them before display. This allows for more precise measurements and lessens the effect of spurious acoustic noise interfering with the measurement, but it also means having to wait longer for the measurement to be ready.

33.4.3 Scale

Name	Description
Range mode	Toggles auto-scaling the vertical axis to the effective range of the impulse response data in the current timeframe. It functions as an automatic zoom alongside the vertical axis, which can be useful for hands-free operation.

33.4.4 Other

Name	Description
Zoom X	Adjusts the horizontal axis zoom factor, which can also be changed by clicking inside the impulse response display itself and rotating the mouse center wheel up and down (scroll in / out), if your mouse has this feature.
Zoom Y-/+	Adjusts the vertical axis zoom factor.
Unit	Change the unit used on the x-axis of the IR graph. Default is "ms". Possible options are: - smp (samples) - ms (milliseconds) - s (seconds)

Name	Description
Width	Change the thickness of the impulse response curve. Default is 1.5 pixels.

A Release Notes

A.1 FLUX:: Analyser 25.01

B System requirements

FLUX:: MiRA:: is built around FLUX::SE's new 2D/3D efficient graphic engine, which employs full GPU-acceleration using an OpenGL-compliant graphics card.

In order to experience the outstanding responsiveness of MiRA, even with a very busy display, and to fully take advantage of the software's analysis capabilities, using a modern nVidia or ATI Radeon graphics card is recommended.

Older and other less efficient graphics cards do not have the required performance and specifications, and offload too much work to the CPU (see below).

The processor is also an important factor, and we recommend using at least and Intel Core 2 Duo, Core i5 or newer architecture processor. AMD processors are also supported, but might exhibit lower performance, as they do not offer the same capabilities and optimizations as Intel CPUs.

B.1 Minimum requirements

- CPU: Intel Core 2 Duo.
- GPU: OpenGL 2.0 or superior compatible, with pixel-shader support.

B.2 Recommended configuration

B.2.1 Apple

CPU/GPU: Apple M1 or better.

B.2.2 Generic Hardware

CPU: Intel Core i5 or better. GPU: AMD/ATI Radeon or nVidia video-card. Intel integrated graphics are not powerful enough and should be avoided.

B.3 Common requirements

A free USB port to connect the iLok key if not using machine authorization

> **Note**
>
> Please check the latest version of vendor-provided, optimized drivers are installed for your graphics card. Generic drivers are generally less up-to-date and may contain bugs or miss optimizations present in drivers specific to your particular model.

B.4 Compatibility

FLUX:: MiRA:: is a 64bit application fully compatible with 64-bit operating systems.

B.4.1 Operating Systems

- PC: Windows 10 or 11
- Apple: macOS versions 10.13 and up (macOS Big Sur, Monterey compliant, Compatible with ARM / Silicon)

B.4.2 Hardware IO support

Any soundcard with a driver compliant with these standards:

- ASIO (Windows).
- Core Audio (macOS).

B.4.3 Software - Sample Push support

SampleGrabber is a 32-bit plug-in compatible using 64-bit double precision internal processing, compatible with 32-bit and 64-bit (via bridge) hosts.

All major native formats (AAX, VST, AU, AAX VENUE) are supported.

B.4.4 Supported formats

- Windows 10
  - VST (2.4)
  - VST3
  - AAX
  - AAX VENUE
- macOS - 10.12 and later
  - VST (2.4)
  - VST3
  - AU
  - AAX

C Scope List

This table lists all the scopes and their availability in the different versions of MiRA::

Name	Category	Included version
Spectrum Analyzer	Spectrum Analysis	MiRA:: Session
Spectrogram	Spectrum Analysis	MiRA:: Session
Nebula stereo	Spectrum Analysis	MiRA:: Session
Wave Scope	Spectrum Analysis	MiRA:: Session
Vector scope	Spectrum Analysis	MiRA:: Session
RMS metering	Level Metering	MiRA:: Session
True peak metering	Level Metering	MiRA:: Session
Loudness metering	Level Metering	MiRA:: Session
Metering History	Level Metering	MiRA:: Studio
Metering statistics	Level Metering	MiRA:: Studio
Nebula 2D	Level Metering	MiRA:: Studio & MiRA:: Live
Nebula 3D	Level Metering	MiRA:: Studio
Signal generator	System measurement	MiRA:: Live
Transfer function	System measurement	MiRA:: Live
Impulse response measurement	System measurement	MiRA:: Live
Leq metering	System measurement	MiRA:: Live

! Important

- MiRA:: Studio and MiRA:: Live have all the scopes of MiRA:: Session
- MiRA:: Ultimate give access to all the scopes

D Factory Layouts List

Name	Scope List	Description	MiRA::Version
Essential	Magnitude Spectrum - Nebula - Vector Scope - RMS Metering - True Peak Metering - Loudness Metering	Basic RTA layout	MiRA::Session
Nebula - Spectrogram	Magnitude Spectrum - Nebula - Spectrogram - RMS Metering	In-depth spectrum analysis	MiRA::Session
RTA	Magnitude Spectrum	Full size RTA spectrum analyzer	MiRA::Session
Sliding compress RTA	Magnitude Spectrum	Full size RTA spectrum analyzer in compressed mode	MiRA::Session
Horizon	Magnitude Spectrum - Nebula - Vector Scope - True Peak Metering	Completed horizontal layout	MiRA::Session
SCOPE	RMS Metering - Vector Scope - Vector Scope - Wave Scope	Big scope with two vector scope, one for spatial, one for transfer function	MiRA::Session
Studio stereo 1	Magnitude Spectrum - Nebula - Vector Scope - RMS Metering - True Peak Metering - Loudness Metering - Metering History	Complete RTA layout	MiRA::Studio

Name	Scope List	Description	MiRA::Version
Studio stereo 2	Spectrogram - Nebula - Vector Scope - RMS Metering - True Peak Metering - Loudness Metering - Metering History	Complete RTA layout with spectrogram instead of RTA spectrum	MiRA::Studio
Studio immersive 1	Spectrogram - Nebula 3D - Vector Scope - RMS Metering - True Peak Metering - Loudness Metering - Metering History	Complete RTA layout for immersive setup	MiRA::Studio
Studio immersive 2	Spectrogram - Nebula 3D - Vector Scope - RMS Metering - True Peak Metering - Loudness Metering - Metering History	Complete RTA layout with spectrogram instead of RTA spectrum for immersive setup	MiRA::Studio
Live show 1	Magnitude Spectrum - Nebula - Vector Scope - RMS Metering - Leq Metering	Layout for live show monitoring	MiRA::Live
Live show 2	Magnitude Spectrum - Nebula - Vector Scope - RMS Metering - Leq Metering - Spectrogram	Variant layout for live show monitoring	MiRA::Live
Live show immersive	Magnitude Spectrum - Nebula - Vector Scope - RMS Metering - Leq Metering - Nebula 3D	Immersive live show monitoring	MiRA::Live
Live show Leq	Leq Metering	Twelve Leq meters	MiRA::Live

Name	Scope List	Description	MiRA::Version
Live - System tuning	Magnitude Spectrum - Transfer Function - Impulse Response	Special layout with multi-channel "FFT" dedicated to system measurement	MiRA::Live
Live - System tuning - no RTA	Transfer Function - Impulse Response	Variant on previous layout without the RTA	MiRA::Live
Live - System tuning - offline	Transfer Function - Impulse Response	A layout design for offline capture processing	MiRA::Live
Live - System tuning - small screen	Transfer Function	A layout designed for smaller screens (14" and below)	MiRA::Live

E Mouse and Keyboard Commands

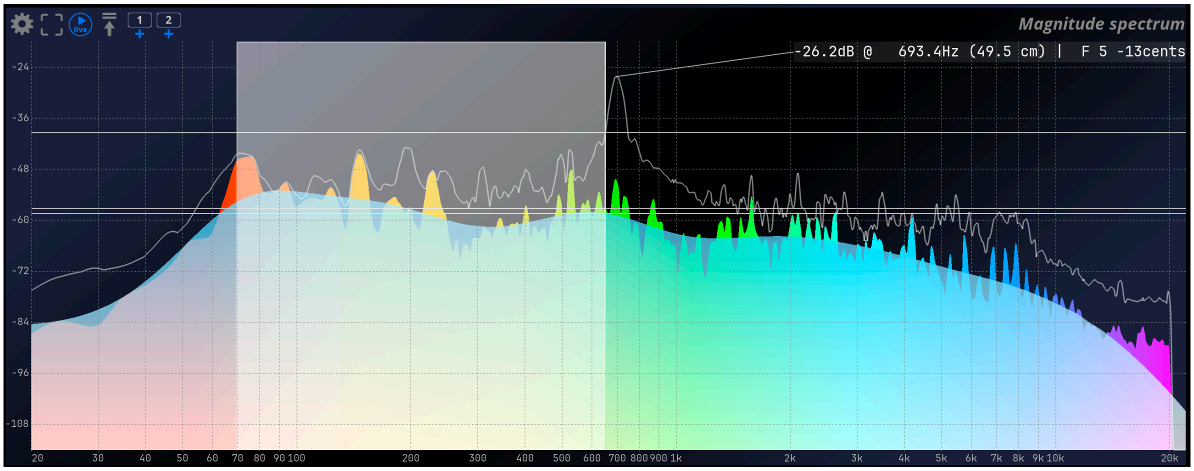
E.1 Mouse commands and conventions

The following mouse click actions are available:

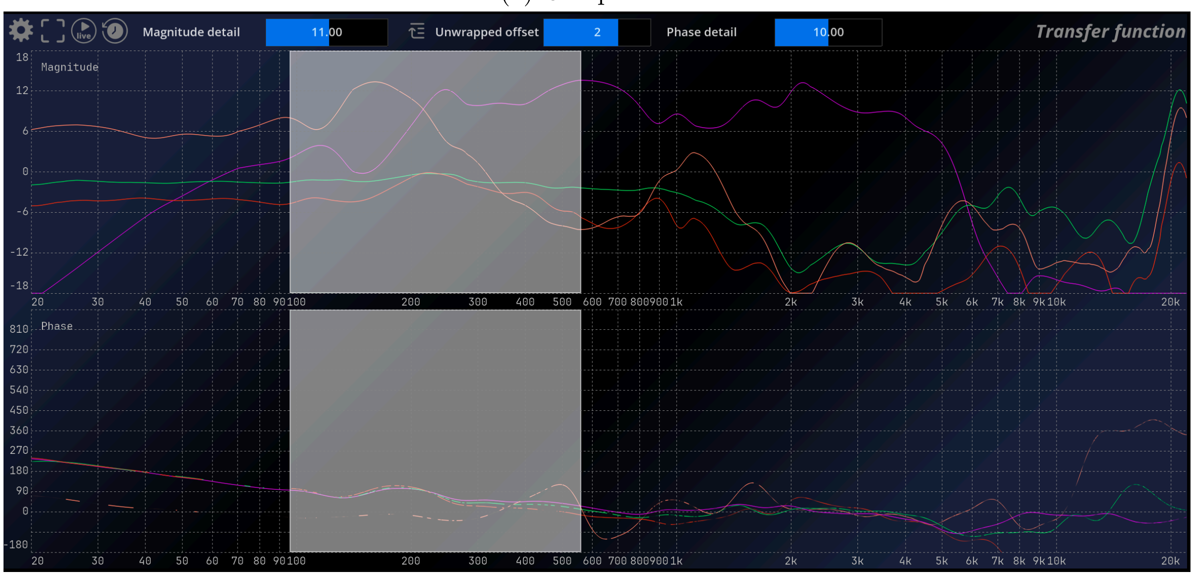
Left-click	Selects the active element.
Right-click	Toggles the display of the corresponding setup menu for the item beneath the current mouse location.
Modifier + click	`Ctrl-click` is equivalent to `right-click`. Inside a setup menu item, `Alt-click` resets the corresponding setting to its default value. `Alt-click` inside an item with a zoom factor greater than one, resets the current zoom to full view (Factor = 1).
Double-click	Double-clicking on an editable control such as a slider or text box enters keyboard entry mode, double-clicking again validates the new value. Double-clicking anywhere inside a panel switches the panel to full-window mode, where the whole application screen is occupied by the corresponding panel; double-clicking a second time reverts to the normal layout.
Click and drag	`Click + drag` inside an item with a zoom factor greater than one shifts the current scale. `Alt + Click + drag` inside an item with a Zoom Factor allows to setup a new zoom according to the defined selection. See Figure E.1 below.
Scroll wheel + click and drag	Turning the middle mouse wheel, if present, affects the current horizontal zoom level of the item under the cursor. Activating the wheel with the middle button simultaneously engaged shifts the current scale when the current zoom factor is greater than one.

E.2 Keyboard shortcuts

E.2.1 MiRA



(a) On spectrum



(b) On IR plots

Figure E.1: Click and drag behavior

Action	Shortcut
About MiRA	*F1*
Main settings	*Cmd + , on macOS, Ctrl + P on Windows*
IO settings	*Alt + , on macOS, Alt + P on Windows*
UI settings	*Cmd + Alt + , on macOS, Cmd + Alt + , on Windows*

E.2.2 File

Action	Shortcut
New workspace	*Ctrl/Cmd + N*
New workspace from factory template	*Alt + Ctrl/Cmd + N*
Open workspace	*Ctrl/Cmd + O*
Save workspace	*Ctrl/Cmd + S*
Save workspace as	*Ctrl/Cmd + Shift + S*
Reload workspace	*Ctrl/Cmd + R*

E.2.3 Edit

Action	Shortcut
Show workspace toolbar	*Ctrl/Cmd + L*
Edit current	*Ctrl/Cmd + Shift + L*
Refresh network connection	*F5*
Toggle generator on/off	*G*
Take offset	*T*
Reset offset	*R*
Mic delay add	*Add*
Mic delay subtract	*Subtract*

E.2.4 View

Action	Shortcut
Load previous layout	Shift + Tab

Action	Shortcut
Load next layout	Tab
Load layout 1	Shift + Alt + 1
Load layout 2	Shift + Alt + 2
Load layout 3	Shift + Alt + 3
Load layout 4	Shift + Alt + 4
Load layout 5	Shift + Alt + 5
Load layout 6	Shift + Alt + 6
Load layout 7	Shift + Alt + 7
Load layout 8	Shift + Alt + 8
Load layout 9	Shift + Alt + 9
Load layout 10	Shift + Alt + 0
Close setup	Escape
Update mouse infos	F6
Always on top	F8
Toggles display of realtime curves	Return

E.2.5 Help

Action	Shortcut
Rebuild GUI	*Shift + Ctrl/Cmd + F5*
Show/hide terminal	*F7*
Show/hide terminal (mini)	*Shift + F7*
Popup terminal	*Ctrl/Cmd + F7*
FOSS - Credits	*F2*
User guide	*Ctrl/Cmd + H*
Show/hide tooltips	*Shift + Ctrl/Cmd + H*

E.2.6 Session

Action	Shortcut
New session	*Shift + Ctrl/Cmd + N*
Import session	*Shift + Ctrl/Cmd + I*
Export session	*Shift + Ctrl/Cmd + E*
Duplicate selected captures in session	*Ctrl/CMD + D*
Duplicate session	*Shift + Ctrl/Cmd + D*
Delete session	*Del/Backspace*

Action	Shortcut
Capture in new session	*Shift + Ctrl/Cmd + Space*
Capture with pink noise in new session	*Shift + Ctrl/Cmd + P*
Capture with sweep in new session	*Shift + Ctrl/Cmd + W*

E.2.7 Capture

Action	Shortcut
New capture	*Space*
New automated capture with pink noise	*Shift + P*
New automated capture with sweep	*Shift + S*
Capture computed curve	*Shift + C*
Invert capture	*Shift + R*
Import capture	*Shift + I*
Export capture audio to .wav file	*Shift + Alt + E*
Delete capture	*Del/Back*
Update capture setting	*Shift + U*
Decrease the delay by one sample	*F3*
Increase the delay by one sample	*F4*
Decrease the delay by ten sample	*Shift + F3*
Increase the delay by ten sample	*Shift + F4*
Find delay	*Shift + F*
Regenerate color	*Shift + G*
Edit notes	*Shift + N*

E.2.8 Generator

Action	Shortcut
Toggle generator on/off	*G*
Next Output	*Shift + O*
Previous output	*Shift + L*

E.2.9 Meters

Action	Shortcut
Refresh all meters	*M*

E.2.10 Metering history

Action	Shortcut
Set Timecode offset	*T*
Reset Timecode offset	*R*

Credits

Project manager and Designer

Gael Martinet

Developed by

Gaël Martinet Lorcán Mc Donagh Nicolas Erard Alexis Gentil Bastien Prevosto Anthony Belard Florie-Anne Lafaye

DSP Algorithms Specialist

Lorcan Mc Donagh

DSP development

Gaël Martinet Lorcán Mc Donagh Maxence Grandidier

Graphic design

Nicolas Philippot, Gaël Martinet

Flux:: Framework development

Gaël Martinet Florie-Anne Lafaye Alexis Gentil Lorcán Mc Donagh Maxence Grandidier Bastien Prevosto Anthony Belard Nicolas Erard Siegfried Hand Antoine Lorence

additional contributions:

Hugo Larin Vincent Carlier Jean-Loup Pecquais Jean Cruypenynck Pablo Arias Samuel Tracol

Flux:: Framework graphic engine

Emmanuel Julien (GS lib), Gaël Martinet

And

and thanks to all fantastic testers...

special thanks to

Alain, Yves, Bruno and Claude for helping to shape our minds over the years

Dolby

Dolby and the double-D symbol are registered trademarks of Dolby Laboratories. Dialogue Intelligence is a trademark of Dolby Laboratories All other trademarks remain the property of their respective owners.

FLUX::

Additional libs

- GS lib Emmanuel Julien, Gael Martinet (Copyright 2013 Emmanuel Julien)
- ThorVG, Copyright (c) 2020 - 2023 notice for the ThorVG Project (see AUTHORS)
- r8brain free - Copyright (c) 2013-2023 Aleksey Vaneev
- LibJpeg - Copyright (c) 1991-2016, Thomas G. Lane, Guido Vollbeding
- libpng :
  - Copyright (c) 1995-2023 The PNG Reference Library Authors.
  - Copyright (c) 2018-2023 Cosmin Truta.
  - Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson.
  - Copyright (c) 1996-1997 Andreas Dilger.
  - Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.
- Freetype 2 - Copyright (c) 2006-2023 by David Turner, Robert Wilhelm, and Werner Lemberg.
- Zlib - Copyright (c) 1995-2022 Jean-loup Gailly and Mark Adler
- bzip2 - Copyright (c) 1996-2010 Julian Seward jseward@bzip.org
- Boost - BSL-1.0
- ni-media - Copyright (c) 2022 Native Instruments
- pfft:
  - Copyright (c) 2004 the University Corporation for Atmospheric Research ("UCAR")
  - Copyright (c) 2013 Julien Pommier (pommier@modartt.com)
  - Copyright (c) 2019 Hayati Ayguen (h_ayguen@web.de)
  - Copyright (c) 2020 Dario Mambro (dario.mambro@gmail.com)
- pybind11 - Copyright (c) 2016 Wenzel Jakob wenzel.jakob@epfl.ch
- rtaudio - Copyright (c) 2001-2021 Gary P. Scavone
- rtmidi - Copyright (c) 2003-2021 Gary P. Scavone
- concurrentqueue - Copyright (c) 2013-2016, Cameron Desrochers.
- readerwriterqueue - Copyright (c) 2013-2021, Cameron Desrochers.
- assimp - Copyright (c) 2006-2021, assimp team
- enet - Copyright (c) 2002-2020 Lee Salzman
- fmt - Copyright (c) 2012 - present, Victor Zverovich and {fmt} contributors
- functions (Erik Rigtorp) - Copyright (c) 2015 Erik Rigtorp erik@rigtorp.se
- inplace_function - BSL-1.0
- gtest - Copyright 2008, Google Inc.
- jsoncpp - Copyright (c) 2007-2010 Baptiste Lepilleur and The JsonCpp Authors
- libcurl - Copyright (c) 1996 - 2023, Daniel Stenberg, daniel@haxx.se, and many contributors
- magic_enum - Copyright (c) 2019 - 2023 Daniil Goncharov
- nanobench - Copyright (c) 2019-2023 Martin Leitner-Ankerl
- nlohmann_json - Copyright (c) 2013-2022 Niels Lohmann

- openddl - Copyright (c) 1999-2022 Eric Lengyel
- openvr - Copyright (c) 2015, Valve Corporation
- poly2tri - Copyright (c) 2009-2018, Poly2Tri Contributors
- pugixml - Copyright (c) 2006-2022 Arseny Kapoulkine
- rapidjson - Copyright (c) 2015 THL A29 Limited, a Tencent company, and Milo Yip.
- shared_recursive_mutex - Copyright (c) 2019 konanM
- sentry-native - Copyright (c) 2019 Sentry (https://sentry.io) and individual contributors.
- sqlite3 - Public Domain License
- sqlite_orm - Copyright (c) 2012-2023 Eugene Zakharov and others
- stb - Copyright (c) 2017 Sean Barrett
- utfcpp - Copyright 2006 Nemanja Trifunovic
- unqlite - Copyright (c) 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019 Symisc Systems, S.U.A.R.L [M.I.A.G + Mrad Chems Eddine chm@symisc.net].

Additional Python modules

- sourcedefender - Copyright (c) 2018-2024 SOURCEdefender. All rights reserved.
- RTTrPM SDK (Blacktraxx) Copyright (c) 2019 CAST Group of Companies Inc
- argh (Andrey Mikhaylenko) - Copyright © 2010—2023 Andrey Mikhaylenko and contributors
- chardet (Mark Pilgrim, Daniel Blanchard)
- charset_normalizer - Copyright (c) 2019 TAHRI Ahmed R.
- docopt (Vladimir Keleshev) - Copyright (c) 2012 Vladimir Keleshev, vladimir@keleshev.com
- dotenv - Copyright (c) 2014, Saurabh Kumar
- ecdsa (Peter Pearson) - Copyright (c) 2010 Brian Warner
- environs (Steven Loria) - Copyright 2022 Steven Loria
- ftputil (Stefan Schwarzer) - Stefan Schwarzer sschwarzer@sschwarzer.net and others contributors
- numpy - Copyright (c) 2005-2023, NumPy Developers.
- pendulum - Copyright (c) 2015 Sébastien Eustace
- psutil (Giampaolo Rodola) - Copyright (c) 2009, Jay Loden, Dave Daeschler, Giampaolo Rodola
- watchdog:
- Copyright 2010 Malthe Borch mborch@gmail.com

- Copyright 2011 Yesudeep Mangalapilly yesudeep@gmail.comCopyright 2012 Google, Inc & contributors.
- tgcrypto - Copyright (c) 2017-2020 Dan https://github.com/delivrance
- idna - Copyright (c) 2013-2022, Kim Davies and contributors. All rights reserved.
- marshmallow - Copyright 2021 Steven Loria and contributors
- packaging - Copyright (c) Donald Stufft and individual contributors.
- paramiko (Jeff Forcier) - Copyright (c) 2003-2008 Robey Pointer robeypointer@gmail.com
- pathtools Copyright (c) 2010 by Yesudeep Mangalapilly
- ptvsd (Microsoft) - Copyright (c) Microsoft Corporation. All rights reserved.
- pyga (Arun K.R.)
- pylightxl - Copyright (c) 2019 Viktor Kis
- pyparsing - Copyright (c) 2003-2019 Paul T. McGuire
- pytzdata - Copyright (c) 2015 Sébastien Eustace
- winshell (Tim Golden) - Copyright (c) 2012 Tim Golden
- six - Copyright (c) 2010-2020 Benjamin Peterson
- Request (Kenneth Reitz)
- sentry_sdk - Copyright (c) 2018 Functional Software, Inc. dba Sentry
- urllib3 - Copyright (c) 2008-2020 Andrey Petrov and contributors
- certifi (Kenneth Reitz)
- anytree (c0fec0de)
- boltons - Copyright (c) 2013, Mahmoud Hashemi
- msgpack (Inada Naoki) - Copyright (c) 2008-2011 INADA Naoki songofacandy@gmail.com
- dateutil - Copyright 2017- Paul Ganssle paul@ganssle.io and dateutil contributors
- PyYAML (Kirill Simonov)

Open source libraries

RTTrPM SDK (Blacktraxx)

ThorVG (replaces crossed-out libraries above) - MIT

r8brain free - MIT

https://github.com/avaneev/r8brain-free-src/blob/master/LICENSE

ooura fft from lmdsp - MIT (or MIT-like)

LibJpeg - Custom (BSD-like)

libpng - PNG Reference Library License version 2

https://sourceforge.net/p/libpng/code/ci/master/tree/LICENSE Copyright (c) 1995-2023 The PNG Reference Library Authors. Copyright (c) 2018-2023 Cosmin Truta. - Copyright (c) 2000-2002, 2004, 2006-2018 Glenn Randers-Pehrson. Copyright (c) 1996-1997 Andreas Dilger. Copyright (c) 1995-1996 Guy Eric Schalnat, Group 42, Inc.

Freetype 2 - FreeType License

https://github.com/freetype/freetype/blob/master/LICENSE.TXT Copyright (C) 2006-2023 by David Turner, Robert Wilhelm, and Werner Lemberg

Zlib - zlib

https://zlib.net/zlib_license.html Copyright (C) 1995-2022 Jean-loup Gailly and Mark Adler

bzip2 - Modified zlib

Copyright (C) 1996-2010 Julian Seward jseward@bzip.org

Boost - BSL-1.0

https://github.com/boostorg/boost/blob/master/LICENSE_1_0.txt

mimalloc - MIT

https://github.com/microsoft/mimalloc/blob/master/LICENSE

ni-media - MIT

https://github.com/NativeInstruments/ni-media/blob/master/LICENSE

pfft

https://github.com/marton78/pffft/blob/master/LICENSE.txt

pybind11

https://github.com/pybind/pybind11/blob/master/LICENSE

rtaudio

https://github.com/thestk/rtaudio/blob/master/LICENSE

rtmidi

https://github.com/thestk/rtmidi/blob/master/LICENSE

concurrentqueue - Simplified BSD

https://github.com/cameron314/concurrentqueue/blob/master/LICENSE.md copyright (c) 2013-2016, Cameron Desrochers. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS " "AS IS" " AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

readerwriterqueue - Simplified BSD

https://github.com/cameron314/readerwriterqueue/blob/master/LICENSE.md Copyright (c) 2013-2021, Cameron Desrochers
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS " "AS IS" " AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

assimp

https://github.com/assimp/assimp/blob/master/LICENSE Copyright (c) 2006-2021, assimp team All rights reserved.

Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of the assimp team, nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission of the assimp team.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS " "AS IS" " AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

enet - MIT

https://github.com/lsalzman/enet/blob/master/LICENSE

fmt

https://github.com/fmtlib/fmt/blob/master/LICENSE.rst Copyright notice Copyright (c) 2012 - present, Victor Zverovich and {fmt} contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

functions (Erik Rigtorp) - MIT

https://github.com/rigtorp/Function/blob/master/LICENSE

inplace_function - BSL-1.0

gtest - " BSD 3-Clause " "New" " or " "Revised" " "

https://github.com/google/googletest/blob/main/LICENSE

jsoncpp - MIT

https://github.com/open-source-parsers/jsoncpp/blob/master/LICENSE Copyright (c) 2007-2010 Baptiste Lepilleur and The JsonCpp Authors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

libcurl - MIT like

https://curl.se/docs/copyright.html Copyright (c) 1996 - 2024, Daniel Stenberg, daniel@haxx.se, and many contributors, see the THANKS file.

All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED " "AS IS" ", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder."

magic_enum - MIT

https://github.com/Neargye/magic_enum/blob/master/LICENSE Copyright (c) 2019 - 2024 Daniil Goncharov

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

nanobench - MIT

https://github.com/martinus/nanobench/blob/master/LICENSE) Copyright (c) 2019-2023 Martin Leitner-Ankerl

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

nlohmann_json - MIT

https://github.com/nlohmann/json/blob/develop/LICENSE.MIT Copyright (c) 2013-2022 Niels Lohmann

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

openddl - MIT

https://github.com/EricLengyel/OpenDDL/blob/main/LICENSE Copyright (c) 1999-2022 Eric Lengyel

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

openvr - " BSD 3-Clause " "New" " or " "Revised" " "

https://github.com/ValveSoftware/openvr/blob/master/LICENSE Copyright (c) 2015, Valve Corporation All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS " "AS IS" " AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES - LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

poly2tri - ” BSD 3-Clause “ “New” ” or “ “Revised” “ ”

pugixml - MIT

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

rapidjson - MIT

https://github.com/Tencent/rapidjson/blob/master/license.txt - "Terms of the MIT License:

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software""), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED""AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

shared_recursive_mutex - MIT

https://github.com/KonanM/shared_recursive_mutex/blob/master/LICENSE Copyright (c) 2019 konanM

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

sentry-native - MIT

https://github.com/getsentry/sentry-native/blob/master/LICENSE Copyright (c) 2019 Sentry (https://sentry.io) and individual contributors. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

sqlite3 - Public Domain

https://www.sqlite.org/copyright.html

sqlite_orm - MIT

https://github.com/fnc12/sqlite_orm/blob/master/COMM-LICENSE Copyright (c) 2012-2023 Eugene Zakharov and others

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

stb - MIT

https://github.com/nothings/stb/blob/master/LICENSE Copyright (c) 2017 Sean Barrett Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software""), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED""AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

utfcpp - BSL-1.0

https://github.com/nemtrif/utfcpp/blob/master/LICENSE "Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization obtaining a copy of the software and accompanying documentation covered by this license (the " "Software" ") to use, reproduce, display, distribute, execute, and transmit the Software, and to prepare derivative works of the Software, and to permit third-parties to whom the Software is furnished to do so, all subject to the following:

The copyright notices in the Software and this entire statement, including the above license grant, this restriction and the following disclaimer, must be included in all copies of the Software, in whole or in part, and all derivative works of the Software, unless such copies or derivative works are solely in the form of machine-executable object code generated by a source language processor.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

unqlite - BSD 2-clause

https://github.com/symisc/unqlite/blob/master/LICENSE " * Copyright (C) 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019 Symisc Systems, S.U.A.R.L [M.I.A.G Mrad Chems Eddine chm@symisc.net]. * All rights reserved. Redistribution and use in source and binary forms, with or without * modification, are permitted provided that the following conditions * are met: * 1. Redistributions of source code must retain the above copyright * notice, this list of conditions and the following disclaimer. * 2. Redistributions in binary form must reproduce the above copyright * notice, this list of conditions and the following disclaimer in the * documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY SYMISC SYSTEMS "AS IS'' AND ANY EXPRESS * OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED * WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR * NON-INFRINGEMENT, ARE DISCLAIMED. IN NO EVENT SHALL SYMISC SYSTEMS * BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF * SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS - OR * BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, * WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE * OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN * IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE." 3rd Party Open Source Python Modules - - - -

argh (Andrey Mikhaylenko) - LGPLv3

https://pypi.org/project/argh/

chardet (Mark Pilgrim, Daniel Blanchard) - LGPL v2.1

https://github.com/chardet/chardet/blob/main/LICENSE

charset_normalizer - MIT

https://github.com/Ousret/charset_normalizer/blob/master/LICENSE Copyright (c) 2019 TAHRI Ahmed R.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

docopt (Vladimir Keleshev) - MIT

https://github.com/docopt/docopt/blob/master/LICENSE-MIT Copyright (c) 2012 Vladimir Keleshev, vladimir@keleshev.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE

FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

dotenv - "BSD 3-Clause""New" " or " "Revised" " "

https://github.com/theskumar/python-dotenv/blob/main/LICENSE Copyright (c) 2014, Saurabh Kumar (python-dotenv), 2013, Ted Tieken (django-dotenv-rw), 2013, Jacob Kaplan-Moss (django-dotenv)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of django-dotenv nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS " "AS IS" " AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

Open source python libraries

ecdsa (Peter Pearson) -

https://github.com/tlsfuzzer/python-ecdsa/blob/master/LICENSE " " "python-ecdsa" " Copyright (c) 2010 Brian Warner

Portions written in 2005 by Peter Pearson and placed in the public domain.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

environs (Steven Loria) - MIT

https://github.com/sloria/environs/blob/master/LICENSE "Copyright Steven Loria

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

ffmpeg-python - Apache 2

https://github.com/kkroening/ffmpeg-python/blob/master/LICENSE " Copyright 2017 Karl Kroening

Licensed under the Apache License, Version 2.0 (the " "License" ") - you may not use this file except in compliance with the License. You may obtain a copy of the License at

```
http://www.apache.org/licenses/LICENSE-2.0
```

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an " "AS IS" " BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License."

ftputil (Stefan Schwarzer) - BSD

https://pypi.org/project/ftputil/ Copyright (c) 2004-2024 Stefan Schwarzer sschwarzer@sschwarzer.net and others contributors

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE"

numpy - "BSD 3-Clause""New" " or " "Revised" " "

https://github.com/numpy/numpy/blob/main/LICENSE.txt Copyright (c) 2005-2024, NumPy Developers. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

```
* Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above
   copyright notice, this list of conditions and the following
   disclaimer in the documentation and/or other materials provided
   with the distribution.

* Neither the name of the NumPy Developers nor the names of any
   contributors may be used to endorse or promote products derived
   from this software without specific prior written permission.
```

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS " "AS IS" " AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

pendulum - MIT

https://github.com/sdispater/pendulum/blob/master/LICENSE Copyright (c) 2015 Sébastien Eustace

psutil (Giampaolo Rodola) - "BSD 3-Clause""New" " or " "Revised" " "

HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

watchdog - Apache 2

https://github.com/gorakhargosh/watchdog/blob/master/LICENSE "Copyright 2010 Malthe Borch mborch@gmail.com Copyright 2011 Yesudeep Mangalapilly yesudeep@gmail.com Copyright 2012 Google, Inc & contributors.

Licensed under the Apache License, Version 2.0 (the " "License" ") - you may not use this file except in compliance with the License. You may obtain a copy of the License at

```
http://www.apache.org/licenses/LICENSE-2.0
```

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an " "AS IS" " BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License."

tgcrypto - LGPL v3

https://github.com/pyrogram/tgcrypto/blob/master/COPYING.lesser Copyright (C) 2017-2020 Dan https://github.com/delivrance

idna - BSD 3

https://github.com/kjd/idna/blob/master/LICENSE.md Copyright (c) 2013-2023, Kim Davies and contributors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS " "AS IS" " AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

marshmallow - MIT

https://github.com/marshmallow-code/marshmallow/blob/dev/LICENSE "Copyright Steven Loria and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

packaging - BSD

https://github.com/pypa/packaging/blob/main/LICENSE Copyright (c) Donald Stufft and individual contributors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE"

paramiko (Jeff Forcier) - LGPL 2.1

https://github.com/paramiko/paramiko/blob/main/LICENSE Copyright (C) 2003-2008 Robey Pointer robeypointer@gmail.com https://raw.githubusercontent.com/paramiko/paramiko/main/LICENS

pathtools - MIT

https://github.com/gorakhargosh/pathtools/blob/master/LICENSE Copyright (C) 2010 by Yesudeep Mangalapilly yesudeep@gmail.com

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

ptvsd (Microsoft) - MIT

https://github.com/microsoft/PTVS/blob/main/LICENSE Copyright (c) Microsoft Corporation. All rights reserved.

Licensed under the Apache License, Version 2.0 (the " "License" ") - you may not use this file except in compliance with the License. You may obtain a copy of the License at

```
http://www.apache.org/licenses/LICENSE-2.0
```

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an " "AS IS" " BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License."

pyga (Arun K.R.) - BSD

https://pypi.org/project/pyga/ Copyright (c) Arun K.R. and individual contributors.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE”

pylightxl - MIT

https://github.com/PydPiper/pylightxl/blob/master/LICENSE Copyright (c) 2019 Viktor Kis

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “ “Software” “), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “ “AS IS”“, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.”

pyparsing - MIT

https://github.com/pyparsing/pyparsing/blob/master/LICENSE Copyright (c) 2003-2019 Paul T. McGuire Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “ “Software” “), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

pytzdata - MIT

https://github.com/sdispater/pytzdata/blob/master/LICENSE Copyright (c) 2015 Sébastien Eustace

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

winshell (Tim Golden) - MIT

https://github.com/tjguk/winshell/blob/master/LICENSE.txt Copyright (c) 2012 Tim Golden

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

six (Copyright (c) 2010-2020 Benjamin Peterson - MIT

https://github.com/benjaminp/six/blob/master/LICENSE Copyright (c) 2010-2020 Benjamin Peterson

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

Request (Kenneth Reitz) - Apache 2

https://github.com/psf/requests/blob/main/LICENSE - modified https://raw.githubusercontent.com/psf/reque

sentry_sdk (Copyright (c) 2018 Sentry) - MIT

https://github.com/getsentry/sentry-python/blob/master/LICENSE Copyright (c) 2018 Functional Software, Inc. dba Sentry

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

urllib3 - MIT

https://github.com/urllib3/urllib3/blob/main/LICENSE.txt Copyright (c) 2008-2020 Andrey Petrov and contributors.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS"", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE."

certifi (Kenneth Reitz) - MPL-2.0

https://github.com/certifi/python-certifi/blob/master/LICENSE "This Source Code Form is subject to the terms of the Mozilla Public License, v. 2.0. If a copy of the MPL was not distributed with this file, You can obtain one at http://mozilla.org/MPL/2.0/."

anytree (c0fec0de) - Apache 2

https://github.com/c0fec0de/anytree/blob/main/LICENSE

boltons - BSD

https://github.com/mahmoud/boltons/blob/master/LICENSE Copyright (c) 2013, Mahmoud Hashemi

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

```
* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above
  copyright notice, this list of conditions and the following
  disclaimer in the documentation and/or other materials provided
  with the distribution.

* The names of the contributors may not be used to endorse or
  promote products derived from this software without specific
  prior written permission.
```

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ""AS IS"" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES ; LOSS OF USE, DATA, OR PROFITS ; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE."

msgpack (Inada Naoki) - Apache 2

https://github.com/msgpack/msgpack-python/blob/main/COPYING Copyright (C) 2008-2011 INADA Naoki songofacandy@gmail.com

Licensed under the Apache License, Version 2.0 (the " "License" ") - you may not use this file except in compliance with the License. You may obtain a copy of the License at

```
http://www.apache.org/licenses/LICENSE-2.0
```

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an " "AS IS" " BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License."

dateutil (Gustavo Niemeyer, Paul Ganssle) - Apache 2

https://github.com/dateutil/dateutil/blob/master/LICENSE "Copyright 2017- Paul Ganssle paul@ganssle.io Copyright 2017- dateutil contributors (see AUTHORS file)

Licensed under the Apache License, Version 2.0 (the " "License" ") - you may not use this file except in compliance with the License. You may obtain a copy of the License at

```
http://www.apache.org/licenses/LICENSE-2.0
```

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an " "AS IS" " BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License."

PyYAML (Kirill Simonov) - MIT

https://pypi.org/project/PyYAML/ Copyright (c) Kirill Simonov.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the " "Software" "), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED " "AS IS" ", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE

FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.0