Synapse Manual

Updated 2025-05-12

Tucker-Davis Technologies
11930 Research Circle
Alachua, FL 32615 USA
Phone: +1.386.462.9622
Fax: +1.386.462.5365

Notices

The latest versions of TDT documents are always online at https://www.tdt.com/docs/

Table of Contents

Gizmo Reference

Synapse Overview

Synapse is the software you'll use to design, manage and collect data from your neurophysiology experiments using System 3 hardware. With Synapse's advanced automation, underlying relational database, and sophisticated hardware interface; the power and flexibility of TDT's proven multi-DSP hardware platform is never more than a few clicks away.

Design

In the design phase of your experiment, Synapse automates all but the highest level set-up tasks. You interact with the software using a streamlined interface where the most commonly modified options are available on easy options tabs. Automated processes combine what Synapse knows about your hardware and what TDT has learned in over 30 years of working closely with researchers like you to deliver smart experiment design.

Let Synapse Remember the Details of Your Hardware for You

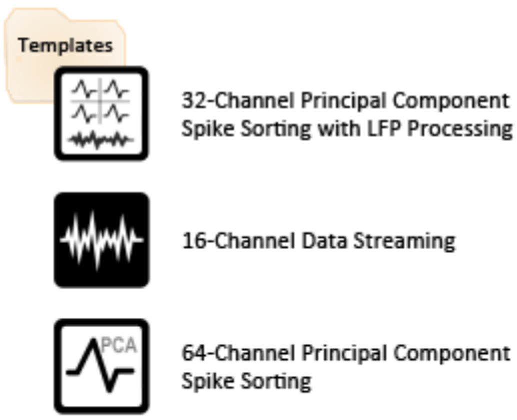
Synapse auto-detects your hardware and uses a hardware abstraction layer (or HAL) between the device specific processes for your equipment and your experiment building selections. When you are configuring recording, data storage, and detection tasks; you don't have to think about the particulars of your hardware system. As you design your experiment, Synapse uses information about your hardware and selections you've already made to show only the relevant choices. You select the parts of the experiment you want and Synapse generates the required code instructions, optimized for your hardware. With only a few mouse clicks, you can build custom experiments and be collecting data in minutes.

Get the Power of Customization without Complexity

Most Synapse users will be able to run out-of-box experiments and acquire data with minimal configuration changes. If you need to do more, there are progressively more detailed levels of design options available. Three powerful paths can be used or combined to support differing levels of flexibility and control.

Experiment Templates

For the fastest design experience, select a pre-made template as the starting point for your experiment. The growing list of TDT designed templates includes multi-channel recording, LFPs, spike sorting, tetrode recordings, and experiments that combine these elements. If the template matches your needs, you can use it as is. For more customized experiments, adjust configuration settings or add gizmo task blocks to make it your own.

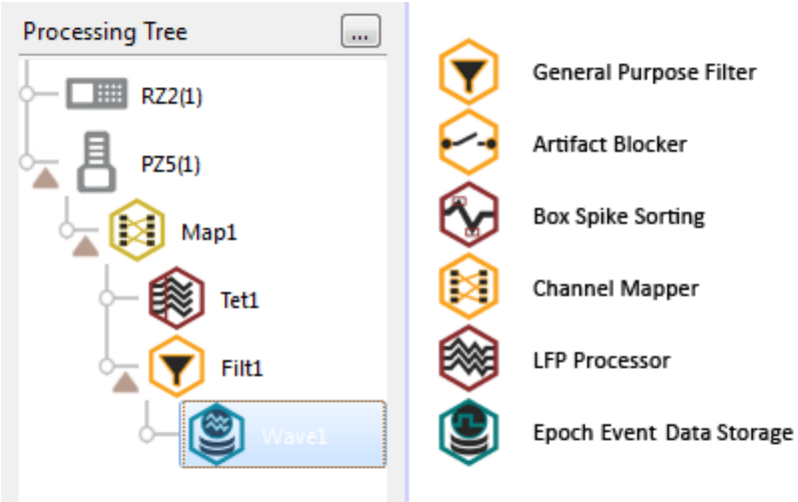


Templates

TDT Gizmos

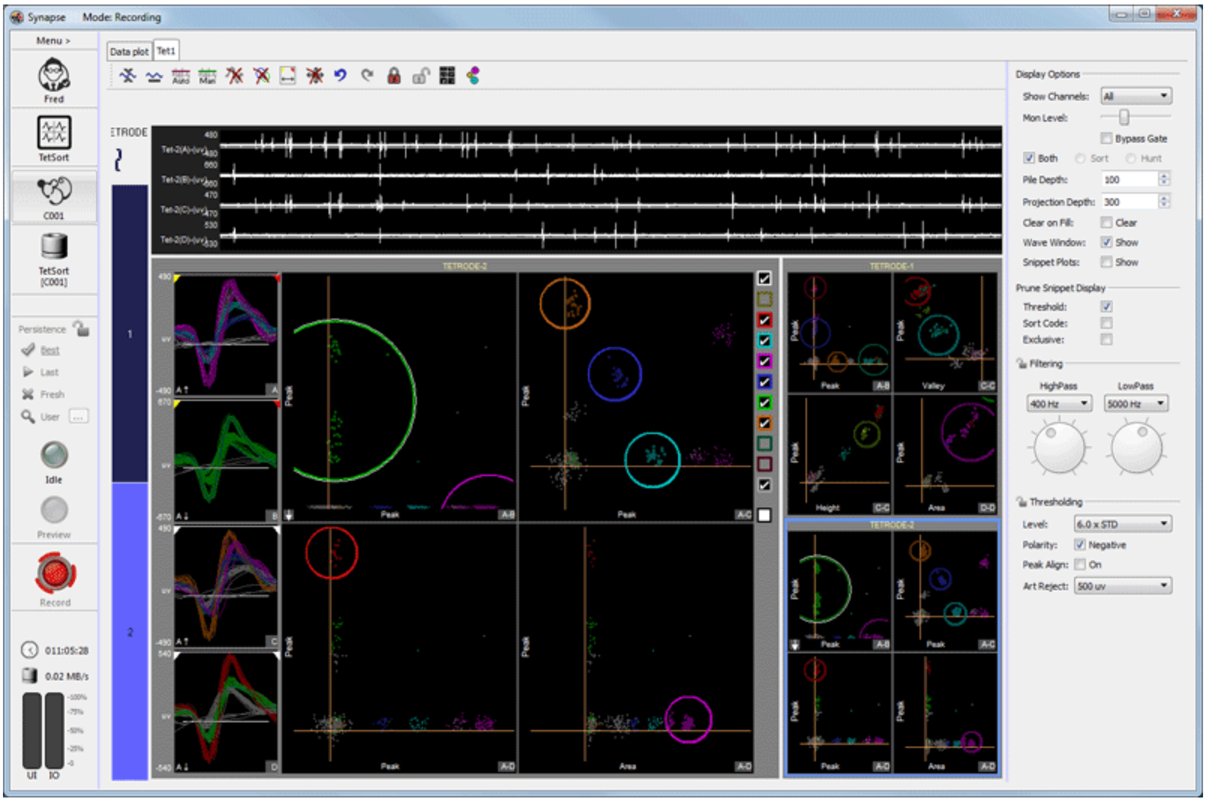
For a balance of flexibility and design speed, select and arrange ready-made building blocks called 'gizmos'. You add them in the order you want each task to occur, for example you would add a filter before a storage task. Synapse adjusts the available options as you work so that only relevant choices are available.

Gizmos are available for a variety of tasks, including reading input signals, filtering, online spike sorting, data storage, channel mapping, stimulation, and much more. Each gizmo can also comprise a group of tasks bundled together for a particular type of experiment, such as online spike sorting and data storage.



TDT Gizmos (partial list)

Many gizmos include runtime interfaces that enable you to make adjustments to your experiment as data is being collected. For example, the Tetrode Processor includes plots to display the four channels of the tetrode as streamed waveforms and as snippets in a pile plot. A specialized 2D feature plot for viewing and selecting different projections is also provided. You can draw clusters or units and apply them from the runtime interface. You can even make changes to the filter settings or threshold levels dynamically and see the changes to the data immediately.

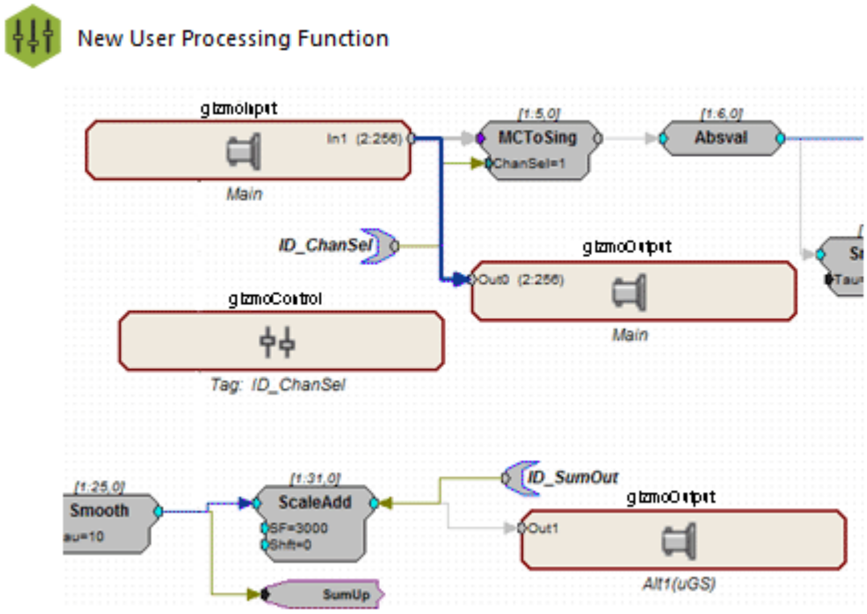


Runtime Interface

User Designed Gizmos

For maximum flexibility and control, build your own custom processing tasks. The gizmo building tools link directly into the signal flow, and include parameter tags or 'hooks' that allow you to control timing, triggering, data storage, and modification of other parameter values dynamically at runtime. For example, you can create a gizmo that runs a novel stimulus protocol, accessing signal parameters directly in the Synapse interface or through a custom application you've developed using TDT provided development tools and MATLAB or Python.

Designing your own gizmos may take a little more time upfront, but once a user gizmo has been created it can be reused in future experiments as easily as the built-ins.



User Designed Gizmo Creation Tools

Get More from Your System with Automation

No matter which method or combination of techniques you use to define your experiment, the Synapse compilation engine determines how to most efficiently utilize your available hardware to run it. One of the reasons System 3 processors are so powerful is that they run multiple DSPs in parallel. Each DSP is capable of running many combinations of tasks, but each task takes a different amount of processing power. Automating the distribution of the tasks that make up your experiment allows you to tap in to the full power of the multiple DSP architecture of the processors in your system from day one and without any special training. With Synapse, you let the computer do the logic based tasks it does so well; giving you more time to do what you do best-- consider big ideas, get creative, and develop insightful conclusions.

Manage

Much of what makes Synapse innovative goes on behind the scenes. Like the automation tools at work as you design your experiments, the relational database is the power behind Synapse's experiment management capabilities. Synapse tags three special categories of information: users, experiments, and subjects; then tracks all runtime settings and any modification made to parameters during each experiment run.

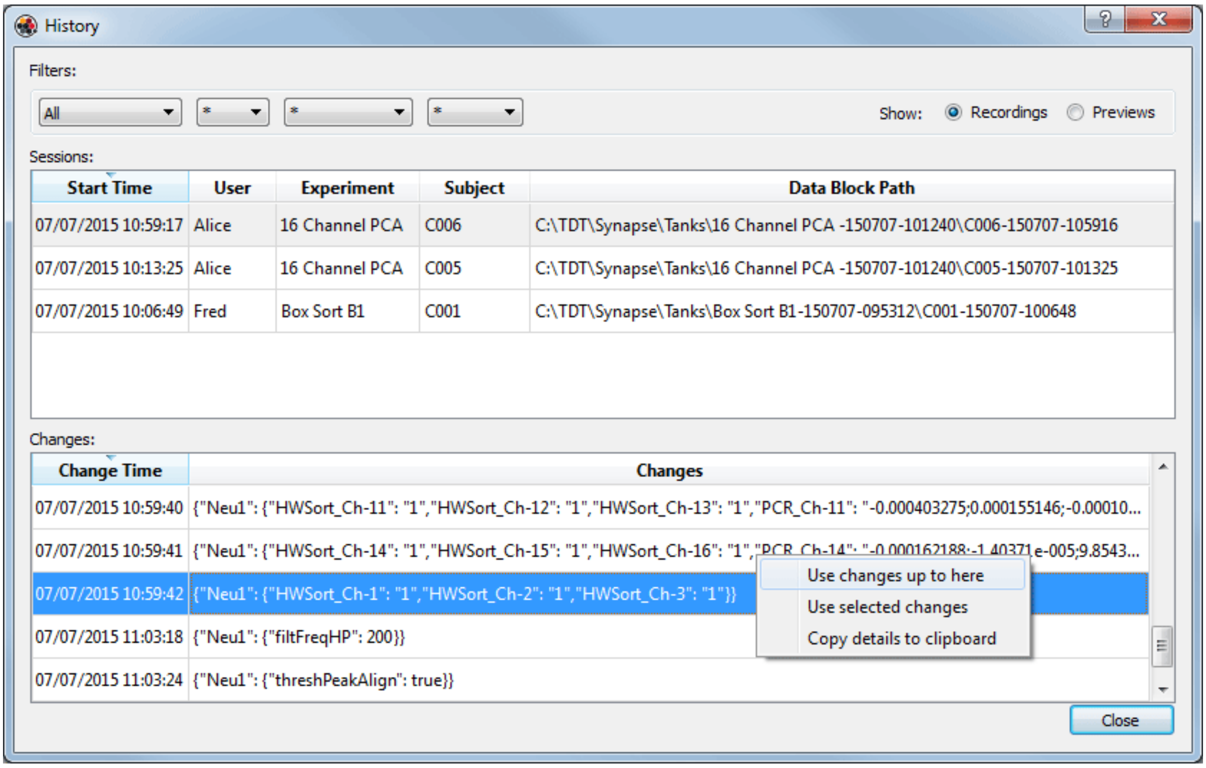
Persistence

You control how Synapse uses the stored runtime configuration data by selecting a Persistence--or way of choosing how settings carryover from user to user, subject to subject, or session to session. Tagging the visual layout and runtime parameters with subject and user information gives your lab several options for customizing the experience for each user. Different users can run a shared experiment with completely different settings. Or individual users can choose to:

- Use the most recent settings for that project or that subject.
- Start over with a fresh interface.
- Start with settings from any previous session.

The Digital Lab Notebook

You can access the complete record of all settings and changes made during each session in the History Window. The relational nature of the database where they're stored enables you to filter sessions in the window by user, experiment, or subject. When you select a session, you can see every change that was made to settings during each run. All of this data is recorded automatically and available at any time.



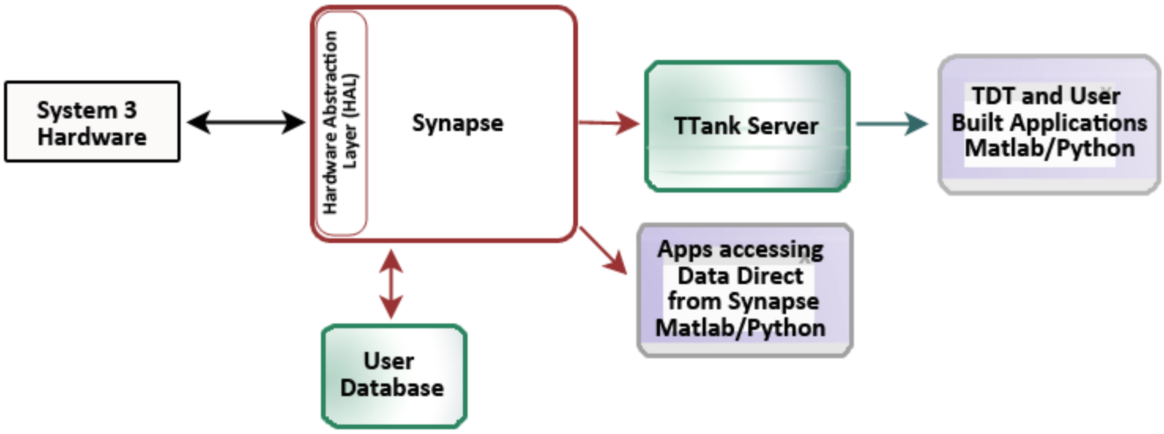
History Window

This window acts as a digital lab notebook where you can see the automatically generated log of all experiments. Select a Session row and the timestamped record of what settings were used or changed in each run (or block) during that session are displayed below. You can even choose a previous set of changes as a launching point for your next session, effectively rewinding your experiment to an earlier state.

The Session rows in the History window can also be used like an export utility, to send experiment data to other applications for visualization and analysis.

Collect

When you are ready to run your experiment, Synapse automatically generates the user interfaces for all gizmos in your configuration. The pre-made user interfaces save time and allow you to make adjustments to your experiment dynamically.



System Communication Flow Diagram

Synapse communicates directly with System 3 hardware for fast, precisely timed operations. With Synapse you can immediately access the data for display while the data is being stored to disk. No system that stores the data before allowing access can compare to Synapse's speed. As data is acquired it is passed to the powerful TTank data server that also got its start in OpenEx. This time-tested data server indexes and stores the data then makes the data available for post hoc visualization and analysis.

Your own custom applications written in MATLAB, Python, or any language that supports ActiveX, can control parameters dynamically at runtime or access the data stored in the Tank format using TDT developed tools.

Synapse not only ensures the integrity of the data you will collect, it also supports integration with existing OpenEx applications such as OpenScope, OpenSorter, and OpenExplorer and maintains compatibility with existing data sets.

Getting Started with Synapse

Before You Begin

Installation

Synapse can be installed from the TDT Installation CD or downloaded from the TDT website as the Synapse and SynapseExtras packages. TDT Drivers should be installed first.

Synapse includes:

- Synapse
- TDT Drivers
- OpenScope
- OpenBrowser
- OpenBridge
- SynapseAPI

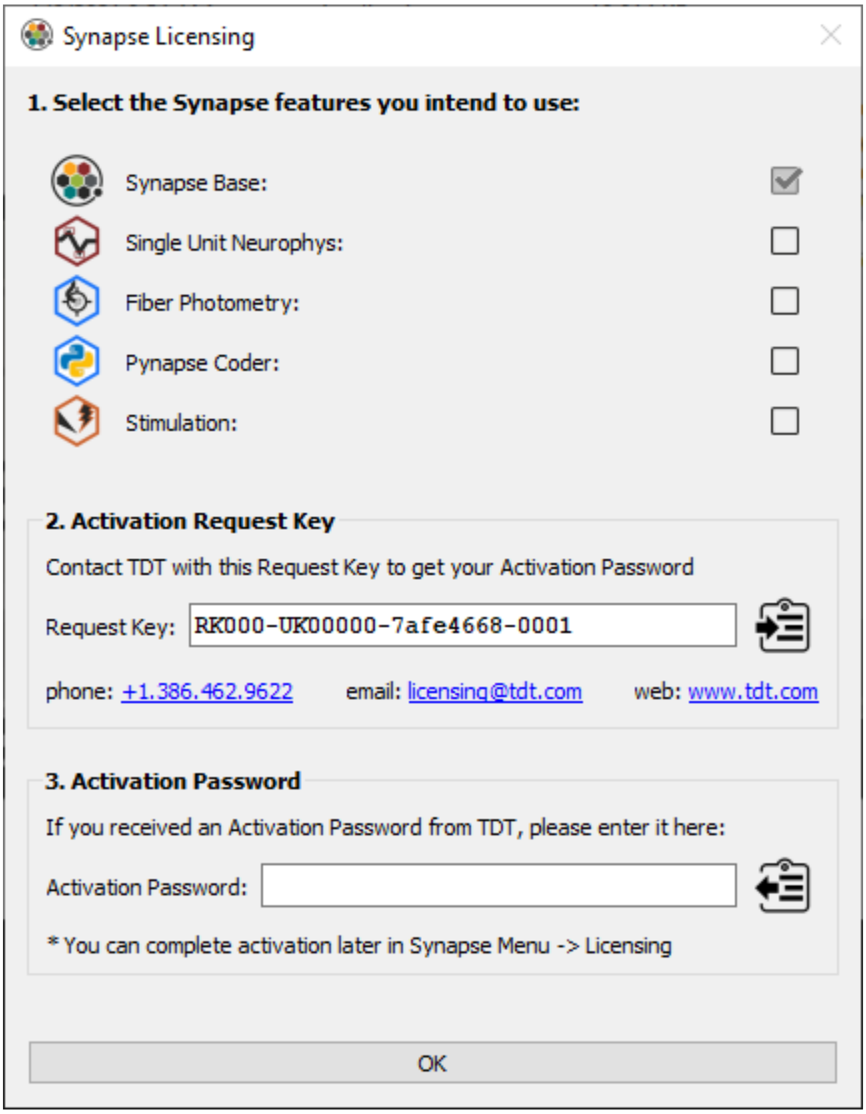
Synapse Extras includes:

- OpenExplorer
- OpenController
- OpenSorter

Synapse Lite is Synapse for LR10 Lab Rat hardware. See Lab Rat User Guide for more information.

> **Package name change**
>
> Prior to v96, 'Synapse Essentials' referred to the base Synapse package above, and 'Synapse Suite' referred to the base Synapse package plus Synapse Extras.

Licensing

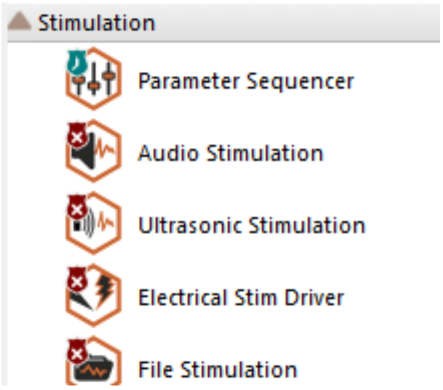


Starting in v96, Synapse requires a license to operate. The gizmos are grouped into categories. This simplifies the gizmo list by pruning gizmos that you don't need, and is more cost effective than purchasing the full Synapse package if you don't need all features.

If you are installing off the USB drive that came with the system, the features that you purchased will automatically be selected. Otherwise, select what you need and contact TDT with the Request Key that it generates. We will send you the Activation Password for your system.

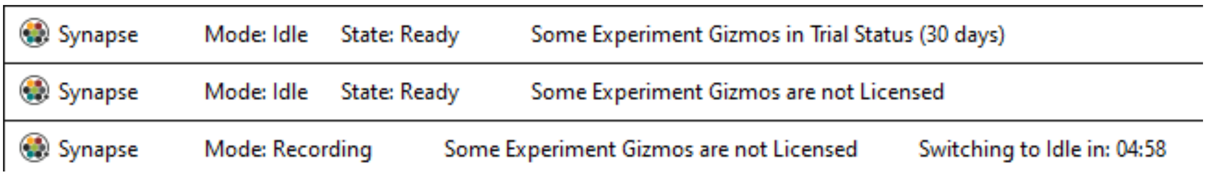
Synapse creates a license file `C:\TDT\Synapse\Synapse.lic` that is specific to the computer that Synapse is installed on. If you need to activate another site, select the features you want on the new computer and contact TDT again with the Request Key from that new computer.

If you want to wait to activate, or try features you haven't purchased, you can select them for a 30 day trial and activate later. The gizmo icons include a badge if they are in trial mode or unlicensed.



Gizmo icons in trial and unlicensed state

The Synapse title bar will tell you if you have experiment gizmos in trial status or gizmos that aren't licensed. If your experiment is using unlicensed gizmos after the 30 day trial expires, runtime will end after 5 minutes.



Synapse title bar shows license status

PC Requirements

The recommended operating systems for all TDT systems is Windows 10.

PC Hardware Requirements:

- 2.0 GHz or faster processor (Intel Core2 Duo or AMD Phenom II processor; 64-bit support recommended)
- 2 GB of RAM (more recommended)
- 1 GB of available hard-disk space for installation (recommended space depends on number of channels and research requirements - contact Tech Support for best options)

- 1080p HD Monitor (1920x1080 display) with OpenGL-compatible graphics card, and 64MB of VRAM (128MB or higher recommended)
- Full height PCIe slot (if connecting to TDT Hardware)

TDT Hardware Requirements

Synapse requires a System 3 Processor and Optibit PC Interface. For best performance, TDT recommends using an RZ Multi-DSP Processor.

See the System 3 Installation Guide for hardware installation and set-up instructions.

File Types

Synapse uses three main user file types:

Extension	Description
.synexpz	Experiment Configuration
.synrig	Hardware Rig
.rcx	Circuit Files for User Gizmos and Legacy Hardware

Synapse can also generate and track user log files associated with a specific experiment, subject, or user. Log files are simple text files that can be read using any text editor.

Data Files

Data is stored using TDT's DataTank format. DataTanks and blocks are treated as folder/file structures. Each new data tank acts as a folder that contains multiple block folders. The files associated with each block are stored within each block folder. They include .tbk, .tsq, .tev, .tdx, and .tin (Synapse experiment information for the block).

> **Tip**
>
> Blocks accessed by OpenScope may contain .tnt files, which are used for annotating data.

Tanks and blocks can be browsed and managed just as you would with other Windows-based folders and files. Individual blocks can be deleted or transferred between tanks using standard Windows methods. However, the underlying file structure for each block should always be maintained. If a block must be moved, move the block folder. Never move or delete an individual file. Blocks and files are named with a consistent naming structure to help keep blocks intact.

Launching Your First Experiment

The quickest way to start collecting data with Synapse is to use a template. After initial system set-up, you can use these ready-to-go experiments to collect data on day one or use them as a starting point for a more customized experiment. This section covers:

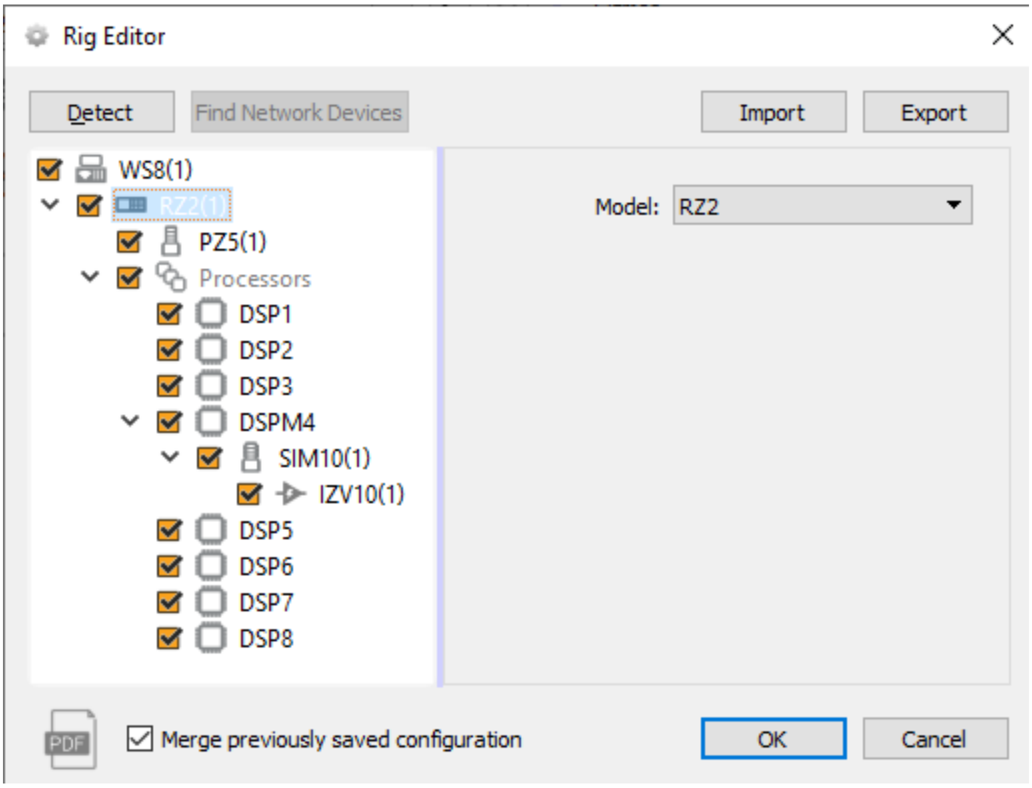
- Editing your Rig
- Creating an Experiment from a Template
- Using the Runtime Interface

Editing the Rig

Before you can run an experiment you must allow Synapse to gather information about the System 3 hardware components in your system. TDT processors come in many configurations that have different capabilities. Once Synapse knows which devices you're using, it will keep track of the device details for you.

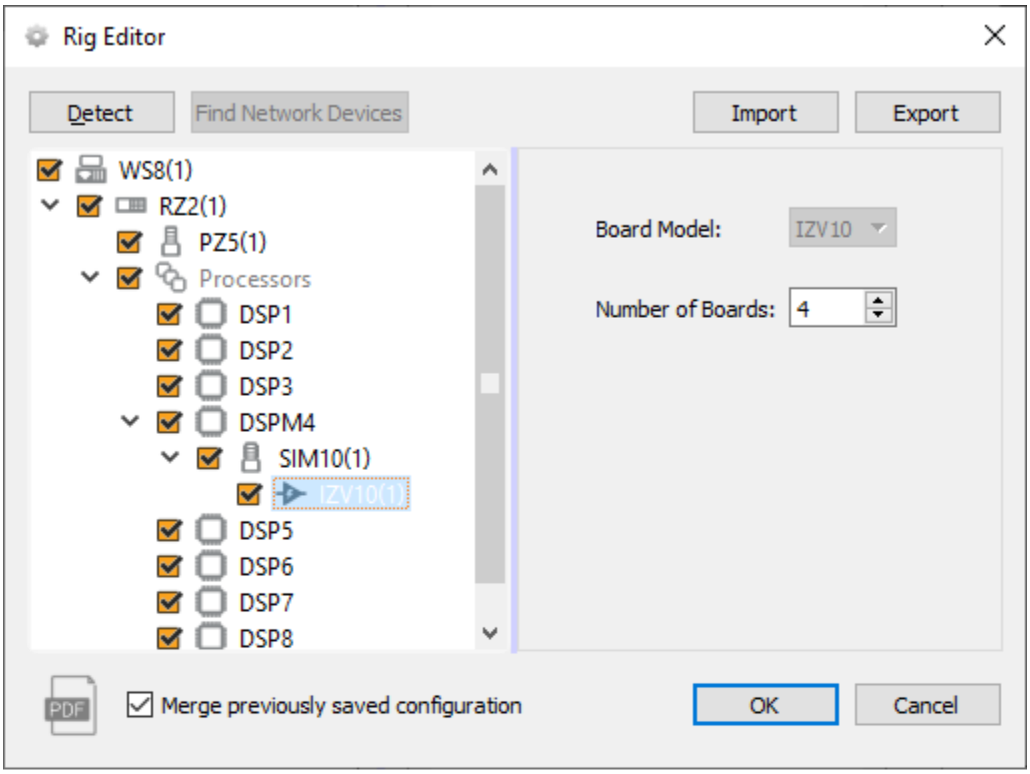
> **Important**
>
> In Synapse, your hardware system is referred to as the 'rig' and it is remembered each time you open the software. The first time you launch the software, the Rig Editor is displayed automatically. You only need to edit the rig if your physical hardware configuration changes.

Configuring the rig starts with letting Synapse detect your system devices. Make sure your TDT hardware is set-up, connected, and powered on before using the **Detect** button to begin configuring the rig.



Rig Editor: RZ2 Selected

Click on a detected device to show configuration settings.



Rig Editor: SIM IZV Selected

> **Important**
>
> There are some devices or configuration settings that Synapse can't automatically detect. For example, IZ2 stimulators don't report their channel count. So you might see an IZ2 in the tree, but it defaults to an IZ2-32. Double-check the peripheral devices in the tree and verify they match your physical hardware.

You can enable or disable devices to control whether they are automatically added to the Processing Tree in new experiments. For example, if you aren't using electrical stimulation on most experiments then you might want to disable the checkbox. Disabled devices can still be added later and enabled manually in the Processing Tree. See Disabled Devices and the Processing Tree for more information.

Adding a Device

The UDP interface, a USB Camera, and RA (Medusa) amplifiers are examples of devices that might be part of your system, but can't be detected by Synapse. These devices can be added to the rig manually using the RZ shortcut menu. Right-click your system's processor device (such as RZ2 or RX8) and select the device from the menu.

Hardware Item	Right-click Option
Medusa PreAmp	Add RAn
USB Camera	Add CAM
UDP Interface	Add UDPRecv & Add UDPSend
BH32 Behavioral Controller	Add BHn

Similarly, if you have an optical quad DSP (DSPQ) with a peripheral device connected to it, such as a Subject Interface, Synapse doesn't know which device it is. Right-click on the DSPQ in the rig and add the device you have connected to that DSP.

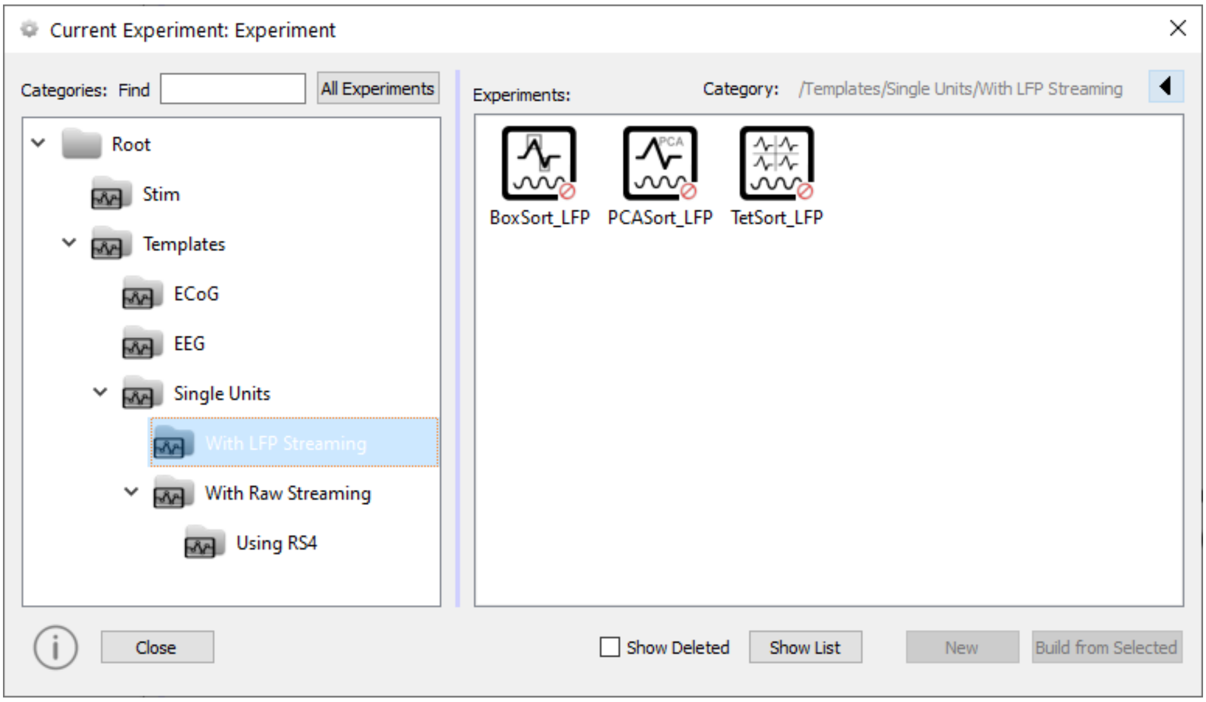
After the rig is initially configured, you won't need to repeat this process in future sessions unless your hardware changes. If you do need to make changes, you can return to the Rig Editor, using the **Edit Rig** command in the main menu.

For more information on working with the rig, see The Rig.

Creating an Experiment from a Template

Templates are pre-built experiments created by TDT to speed up experiment creation. Each Synapse template is a basic working experiment that can be run as configured or modified to meet your needs.

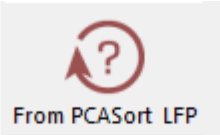
You can access any saved experiments by clicking the **Experiment** button on the command bar, then clicking **More**. Templates are stored in special category folders within the Templates folder.



Current Experiment Window

The Current Experiment or Experiment Selection window is similar to a standard Windows Explorer window with folders, or categories, on the left and experiments in the category on the right.

The rest of this section will take a look at the following template:
Templates | Single Units | With LFP Streaming | PCAsort_LFP



Template files are locked to ensure you will always have an unaltered set in their original state. Select the desired experiment template and click the **Build From Selected** button, to create an editable copy.

The Rig and the HAL

Synapse has two ways of remembering information about your hardware.

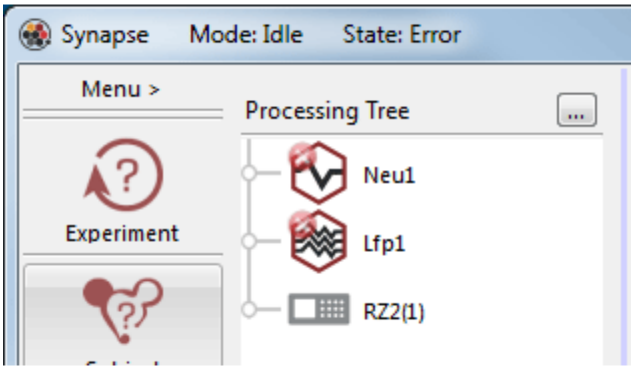
The Rig HAL

The Rig Hardware Abstraction Layer (HAL) is the collection of hardware information that ***stays with your copy of Synapse***. It is used to execute your experiments.

The Experiment HAL

The Experiment Hardware Abstraction Layer (HAL) is a collection of hardware information that is ***stored with the experiment***. It remembers the hardware used to create the experiment.

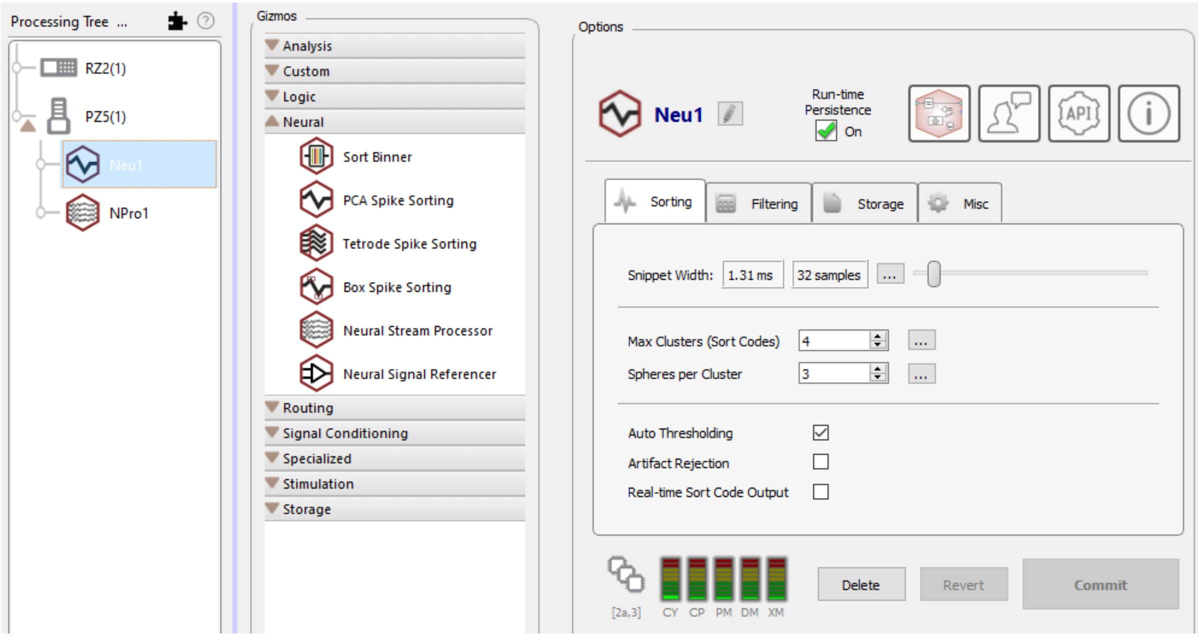
Each template contains HAL information about the system used to design it. When the template is launched, Synapse tries to adapt it to run on your rig. If it is unable to do so, alert symbols are added to the problem elements in the Processing Tree.



Processing Tree with Error Alerts

In the illustration above, the necessary PZn wasn't enabled in the rig where the template was opened. The problem is easily corrected by enabling the PZn.

Viewing the New Experiment

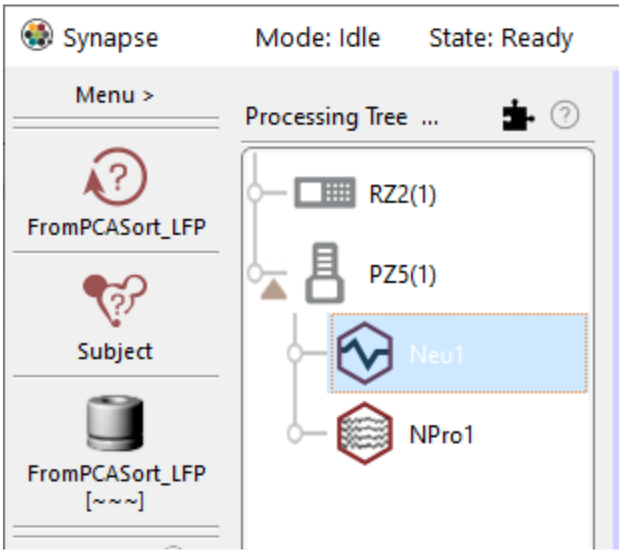


PCA Spike Sorting Gizmo Options in Synapse Window

In the new experiment, two task blocks called 'gizmos' are added to the Processing Tree beneath the PZ amplifier: one for LFP filtering (NPro1) and another for PCA spike sorting (Neu1). The Processing Tree represents the path of data flow and in the example above the hierarchy shows that the LFPs and Single Units are being acquired in parallel from the same signal source (PZ5). When a gizmo is selected in the tree, its configuration options are displayed in the Options area to the far right. For information on modifying gizmo options, see the corresponding reference in the Gizmo Reference).

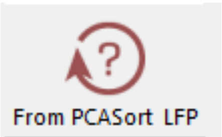
Naming the Experiment

Before you name the experiment, take a quick look at the Synapse interface.

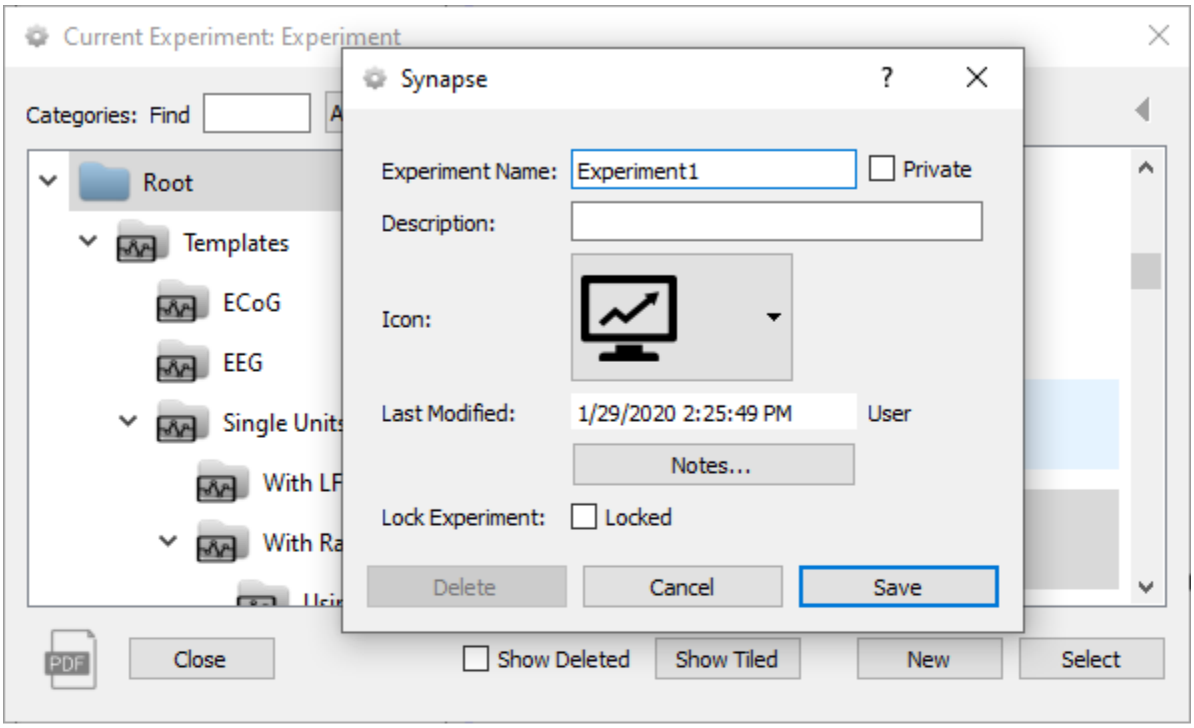


Synapse Designtime Interface

Notice the large buttons at the top of the command bar, seen on the left side in the illustration above. The buttons that are initially displayed in red and are switched to black as each area is configured. Until configuration is complete the experiment **Record** button is unavailable.

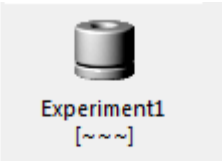


Before you can begin collecting data, the new experiment needs to be named and saved. Click the **Experiment (From PCASort LFP)** button and click **Save as**.



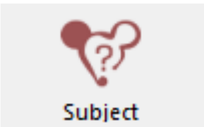
Current Experiment Window and New Experiment Dialog

In the dialog box you can enter an experiment name, description, or add notes. The experiment is saved under the **Root** directory. You can add and move categories and experiments using right-click menus.

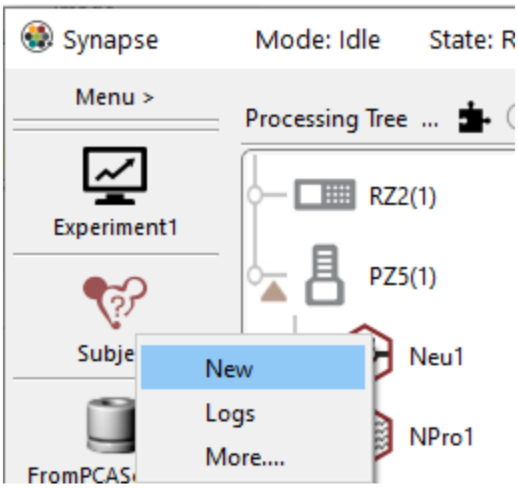


When you return to the main designtime interface, the experiment has been saved, a new tank name is displayed under the tank icon, and the **Record** button is available.

The experiment name is one of three special categories of information that Synapse tags and uses in its relational database to index and track designtime and runtime settings and any modification made to parameters during each experiment run. The other two categories are users and subjects. Which of these buttons is displayed is controlled by Synapse preferences. In the default configuration only the experiment and subject are displayed and both must be configured before recording.



To add a subject, click the **Subject** button in the command bar, then click **New**.



Subject Menu

You'll need to enter a name in the **Subject Name** field. You can also enter a description, password, or notes and choose an icon. When you're done adding information, click **Save**.

The steps to add users and subjects are much the same. For more information about the user, subject, and experiment features, see Managing Users and Subjects.

Using the Runtime Interface

When an experiment is fully configured and saved, the **Preview** and **Record** buttons are enabled. In preview mode you can display data, adjust plots, and change runtime settings without any of the data being permanently stored to the data tank. This is particularly useful for tasks like spike sorting, where you might want to establish the sorting parameters before collecting data. For more straightforward tasks, like recording streamed data, you might choose to skip preview and go straight to record mode.

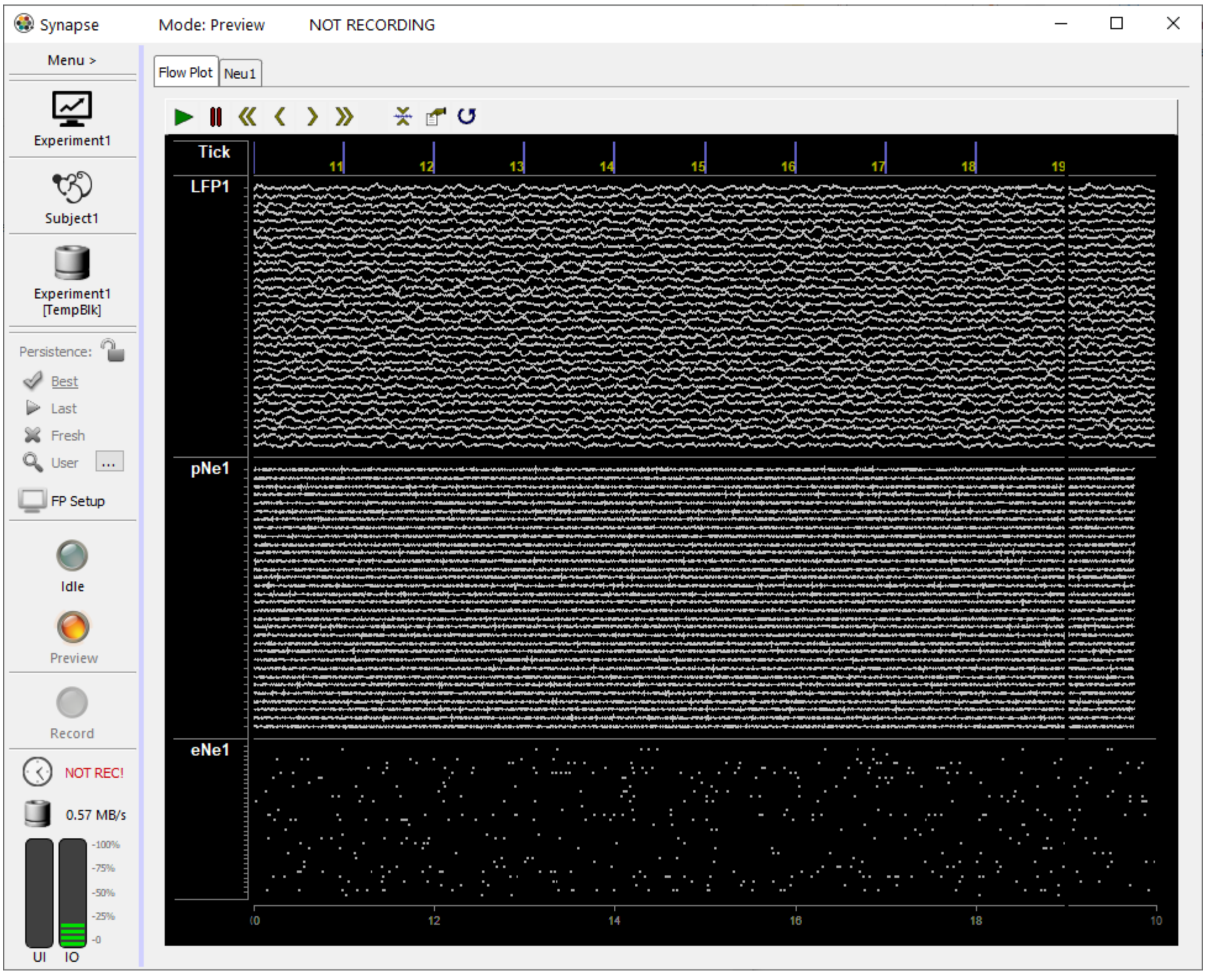
Mode Buttons

Mode	Description
Idle	Devices are not loaded and are not running
Preview	Data is acquired, but deleted after the recording ends
Record	Data is acquired and stored to the data tank permanently

The Runtime Window

The runtime window includes tabs with the main data plot and runtime controls for each gizmo. The data displayed is pulled directly from the hardware and sent to the display in parallel with data storage.

The Flow Plot includes a plot for each type of data being stored. Each plot is automatically configured according to the type of data, for example: snippet, streamed waveforms, or epoch events. You might need to scale the plots to display the waveforms appropriately. This window is explained in more detail in Runtime.



Main Plot at Runtime - Streamed LFPs, Plot Decimated Waveforms, and Snippets

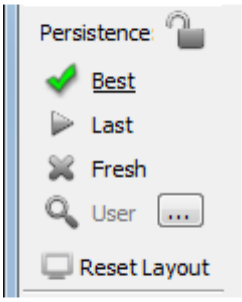
The second tab in the template is from the PCA Spike Sorting gizmo and is an interactive display with plots for cluster cutting and provides runtime access to many of the configuration setting, such as filter values, display options and even the sorting algorithm. You can find more information on this interface in PCA Spike Sorting.

To make sure you don't lose your cluster definitions and other settings between runs, Synapse will remember them according to your persistence setting.

Persistence

By default, Synapse saves the state of all experiment variables, including filter settings, threshold values, and cluster definitions in its relational database during each recording session. Any change made to a setting is logged in the database with the current subject and experiment. This database supports several useful features including persistence and history browsing, filtering, and export features.

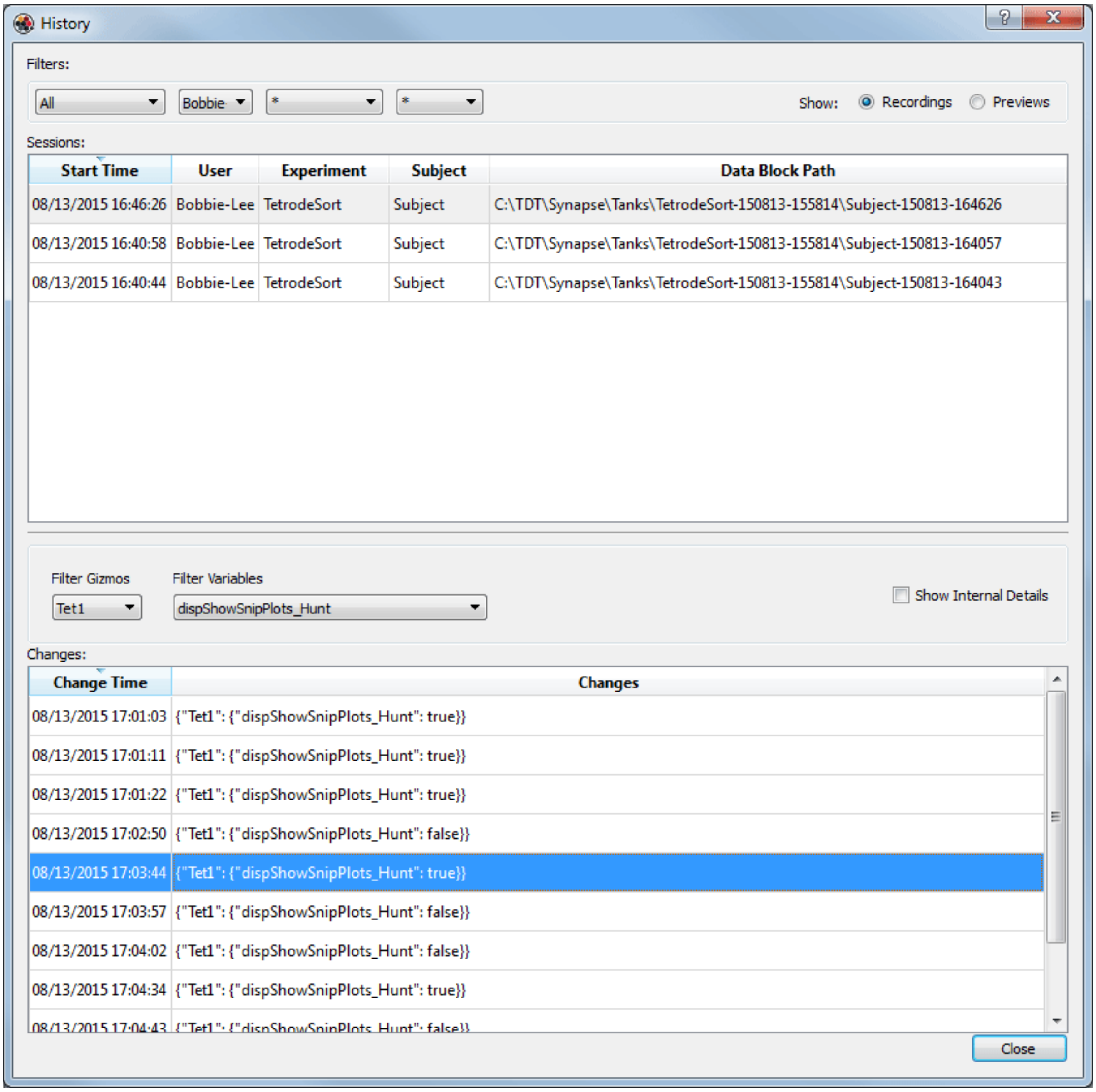
In the runtime window, 'persistence' refers specifically to how past experiment states are applied the next time the experiment is run. You can choose this behavior using the Persistence choices on the command bar.



Mode	Description
Lock	Locks selected persistence option
Best	Uses the last settings of any runtime controls for the current experiment and user/subject
Last	Uses the last settings of any runtime controls (regardless of subject, user, or experiment)
Fresh	Uses the settings in the designtime Options area and clears any past runtime settings
User	Choose settings from a previous recording to apply to next recording. More on that below.

Unless you lock the choice, persistence returns to the **Best** default behavior after each recording session.

History



History Window

This window includes a variety of filtering mechanisms to help you quickly find the data, experiment state, or specific setting value you are looking for. Using shortcut (right-click) menus you can select and return to start and end states for each session or any change state during the session. The top section of the window also displays the path and location for the data recorded during that session. You can begin working with the data set immediately from this window, using the same shortcut (right-click) menus.

Window Layout

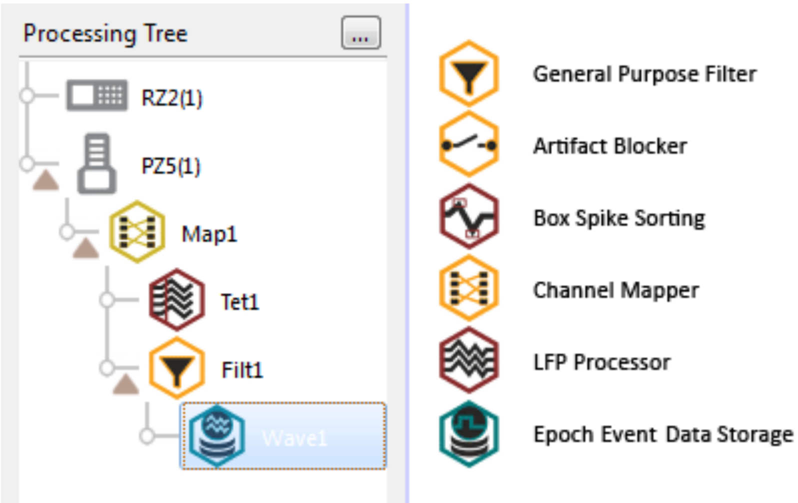
Synapse can also remember information about tab layout. You can drag tabs to float them in new windows or right-click the tab to control placement inside the main window. Information about the window layout is specific to the user and is separate from persistence information. In Idle mode, you can setup the layout using the **RT Layout** button on the command bar.

The Data Tank

TDT's TTank data server indexes and stores recorded data then makes the data available for post hoc visualization and analysis. By default, Synapse names data tanks (a grouping of recordings) automatically based on the experiment's name. Blocks (single recordings) are automatically named based on the subject ID for that recording session. These default preferences are ideal for labs that run an experiment on multiple subjects then move on to another experiment. If your lab does things differently, such as running experiments on the same subject to compare results over time you can change the Synapse preferences to organize the data is different ways. For more information, see Managing Data for Your Lab.

Using Gizmos to Build an Experiment

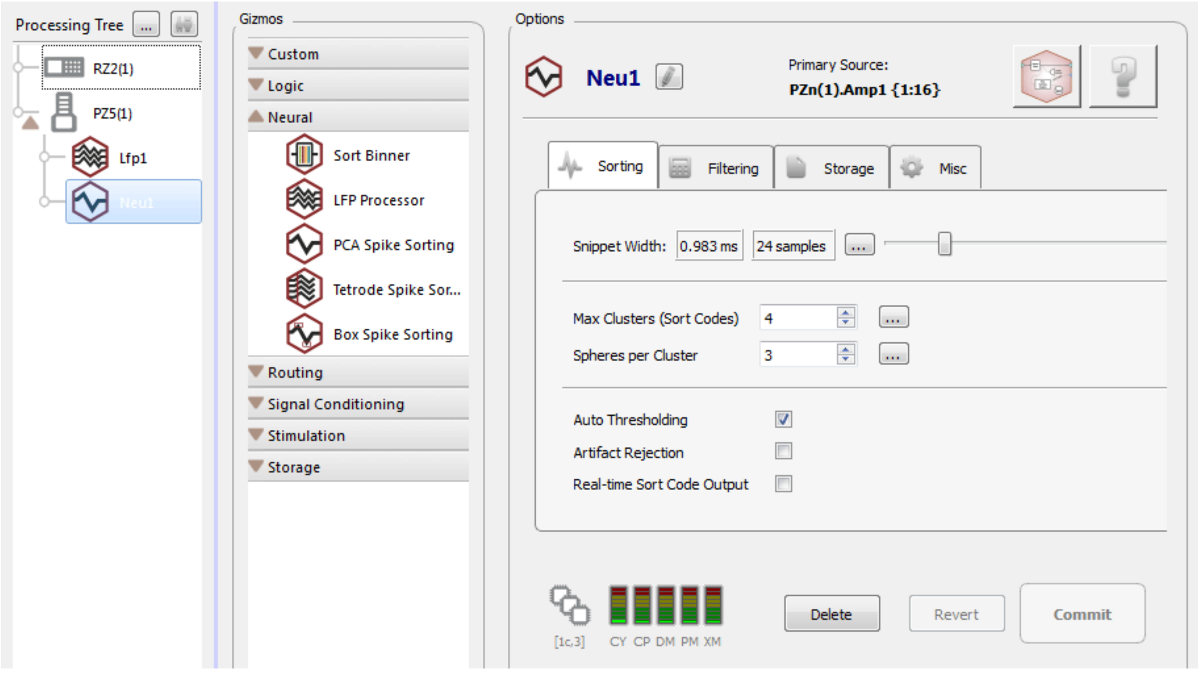
Synapse experiments are a collection of building blocks, called "gizmos", that you can combine to assemble your experiments. Gizmos are available for a variety of tasks; including reading input signals, filtering, online spike sorting, data storage, channel mapping and much more. Each gizmo can be a single task or comprise a group of tasks bundled together for a particular type of experiment, such as online spike sorting and data storage.



Synapse Gizmos (partial list)

Synapse's designtime experiment interface is streamlined to show what you need when you need it. As you make selections, relevant options are displayed. In addition to the command bar, the interface is divided into four areas.

- Processing Tree: Your selections here determine what's displayed in other areas.
- Gizmos: Gizmos that can be added to the selected item in the Processing Tree, organized by type.
- Options: Configuration options for the item selected in the Processing Tree.



Designtime Window

The Processing Tree

The Processing Tree shows the parts of your experiment in a hierarchical diagram. Each experiment can have many parts or branches that form your experimental "machine." It's easiest to identify each branch by looking at the point where signals are input or output.

RZ processors and other devices that represent the starting point (or potential starting point) for a branch form the trunk of the tree. Biological signals are typically acquired by an amplifier, so amplifiers also appears on the trunk of the tree. It is important to note that most acquired signals pass through the RZ for processing and storage, but that does not necessarily mean all acquisition gizmos should be added to the RZ branch. They should be attached to the device that serves as the primary input source for a particular signal set.

If a device in your system does not show up automatically in the Processing Tree, you will need to edit the Rig. See The Rig.

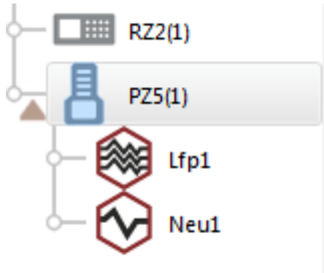
Gizmos for tasks such as filtering, signal processing, and data storage, form the branches of the tree diagram. They are added to the device or gizmo that will be used to input or output the signals associated with the particular task.

To add a gizmo:

- After you have selected an item in the Processing Tree, double-click the gizmo in the Gizmos list or drag it into place.

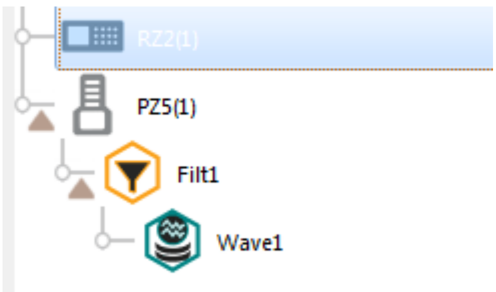
Where to Add a Gizmo

Neural processes, like spike detection, are acquired using an electrode and headstage connected to an amplifier, so they are added to the PZ Amplifier in the tree.



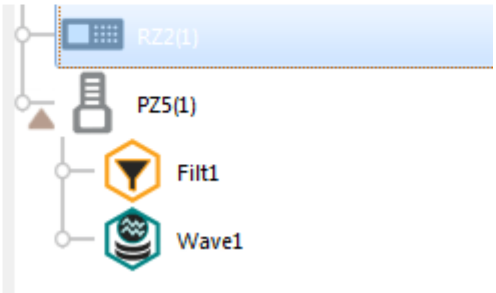
Spike Sorting Gizmos added to PZ5 Amp

Multiple tasks can be added to any branch, in parallel or in a chain, with tasks ordered according to data or signal flow.



Filter and Store Gizmos in a Chain

A stream store gizmo is added to a filter. So, filtered waveforms are stored.



Filter and Store Gizmos in Parallel

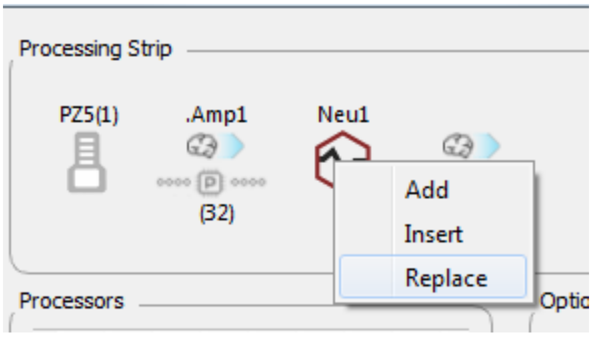
The store and filter are added in parallel. So, raw waveforms directly from the amplifier are stored. The stored waveforms have not passed through the filter.

At each step of the design process, Synapse uses information about your hardware and selections you've already made to show only allowed choices.

Working with Gizmos

Drag and Drop

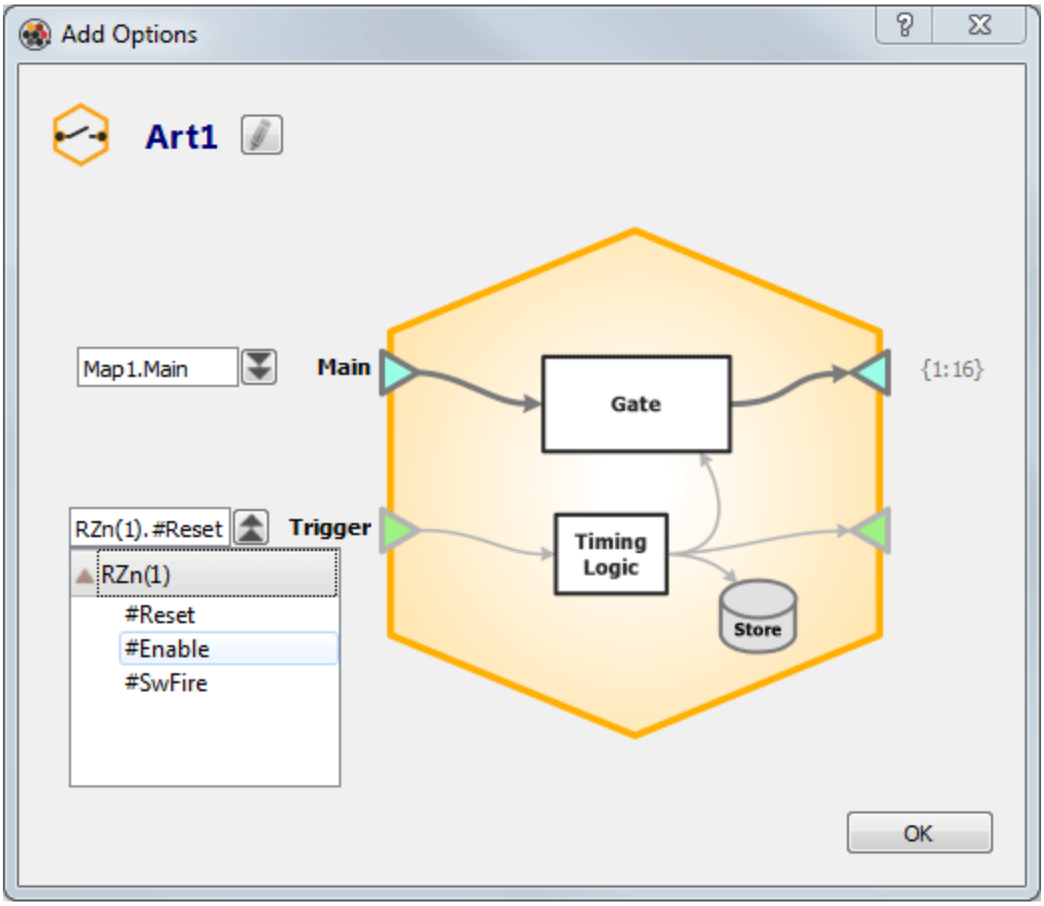
You can drag gizmos from the Processor List or within the Processing Tree to add or reorder them. You can also drag a gizmo to the Processing Strip (enabled in the Synapse Preferences) to replace a gizmo in an existing branch displayed there. When you drop the gizmo, a shortcut menu appears to allow you to add, replace or insert. The Processing Tree will update to reflect these changes.



Processing Strip Menu

Data Types

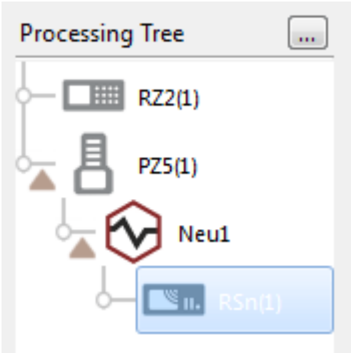
If a gizmo that requires you to select a source is added to the Processing Tree, the gizmo's block diagram is displayed. In the drop-down menus for each input the available list of inputs is arranged and filtered by accepted data types (multi-channel, single-channel, float, integer, TTL).



Gizmo Block Diagram with Trigger Source Selector

Adding a Device to a Gizmo

Some devices, such as the RS4 data streamer, initially appears at the trunk of the Processing Tree and must be moved to the end of a branch. You can drag a device to any gizmo with a multi-channel signal output or change its primary source in the Options area.



Processing Tree with RS4

Two-Sample Delays

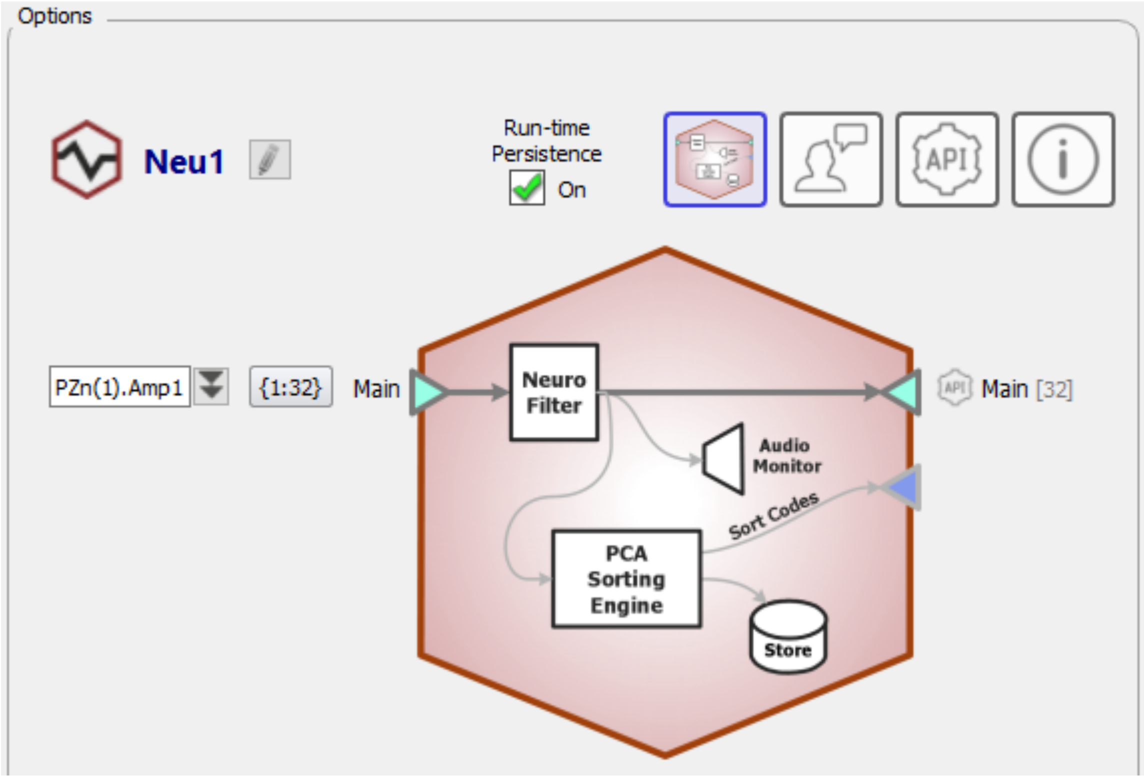
Each branch in the processing tree represents a signal flow or processing path. Each gizmo or device in a branch adds a two sample delay to that path. The path pictured above, from the PZ5 to the RSn, will have a four sample delay. You'll need to be aware of these small delays for experiments with complex processing paths that include many gizmos, custom user gizmos, or where timing is critical. See Creating User Gizmos for more information on user gizmos.

The Options Area

When a gizmo is selected (highlighted) in the tree, its configuration options are displayed in the Options area to the far right. This area also includes a block diagram that provides a logical view of the processes in the gizmo.

The diagram is displayed by clicking the **Block Diagram** button.

The Block Diagram



Block Diagram with Signal Source Selector

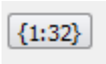
The block diagram in the Options area of the designtime window provides a quick reference for what inputs, outputs, and data is stored by the gizmo. The input source or number of channels and external triggers can also be modified here. The drop-down menus are populated with relevant and valid options.

Gizmo Naming Scheme

Synapse automatically generates and uses consistent naming wherever possible. For example, the gizmo name is generated automatically for you and is based on the type of gizmo selected as well as the index number, so you can use multiple gizmos of the same type in one experiment. In the illustration above, the name "Neu1" is automatically generated for the PCA Spike Sorting gizmo, added from the Neural group of gizmos, and the name and gizmo index are used to name the corresponding data store.

TDT recommends using the auto-generated naming schemes to help make experiments and data consistent across users and subjects.

Channel Counts



The **Channel** button displays the first channel number and the number of channels (or count), indicating the current range of channels included in the primary source input to the gizmo. By default, all available channels are selected.

Clicking the button displays the Select Channel Range dialog box where you can change the range to select a subset of available channels. For example: to sort and store spikes for channels 33-48, clear the **All** check box and enter: First: "33" and Count: "16."

Automatic Propagation of Changes

Changes to channel count in the data stream automatically propagate through the tree branch to attached gizmos.

For information on modifying gizmo options, see the corresponding gizmo in the Gizmo Reference.

Global Buttons

Located below the options tabs, the **Delete**, **Revert**, and **Commit** buttons are applied across all tabs.

The experiment is saved each time you make or "Commit" a change. If you need to roll back a change in the experiment design after it has been committed, you can do so in the Experiments Revision Log. See Experiment Changes and the Revision Log for more information.

Managing Data for Your Lab

Data Tanks and Blocks

After you've created an experiment, the next step is to consider how and where the data will be saved. Synapse preferences allow you to choose where tanks are stored. By default the tank path is: `C:\TDT\Synapse\Tanks`

Tank and Block Naming

Synapse provides a structured but flexible automated solution for tank and block naming based on your preferences. To understand how this works, you first need to understand that Synapse recognizes experiments and subjects as key categories of information that play a special role in managing data storage and retrieval.

Large buttons for these key categories are positioned prominently at the top of the command bar. These buttons are both functional for configuring experiments and serve to display current selections. Below the key category buttons, an icon displays a truncated version of the current tank and block names, which are generated from the experiment and subject names. The full names also include date and time information.

By default, Synapse names data tanks automatically based on experiment name and the start time of the first recording {ExperimentName}-{yymmdd}-{hhmmss}. Blocks of data are named based on subject {SubjectName}-{yymmdd}-{hhmmss} for each recording session and the start time.

For example, an experiment named "LFPbase" run for the first time on October 8, 2015 at 10:51:34 is named:
LFPbase-151008-105134

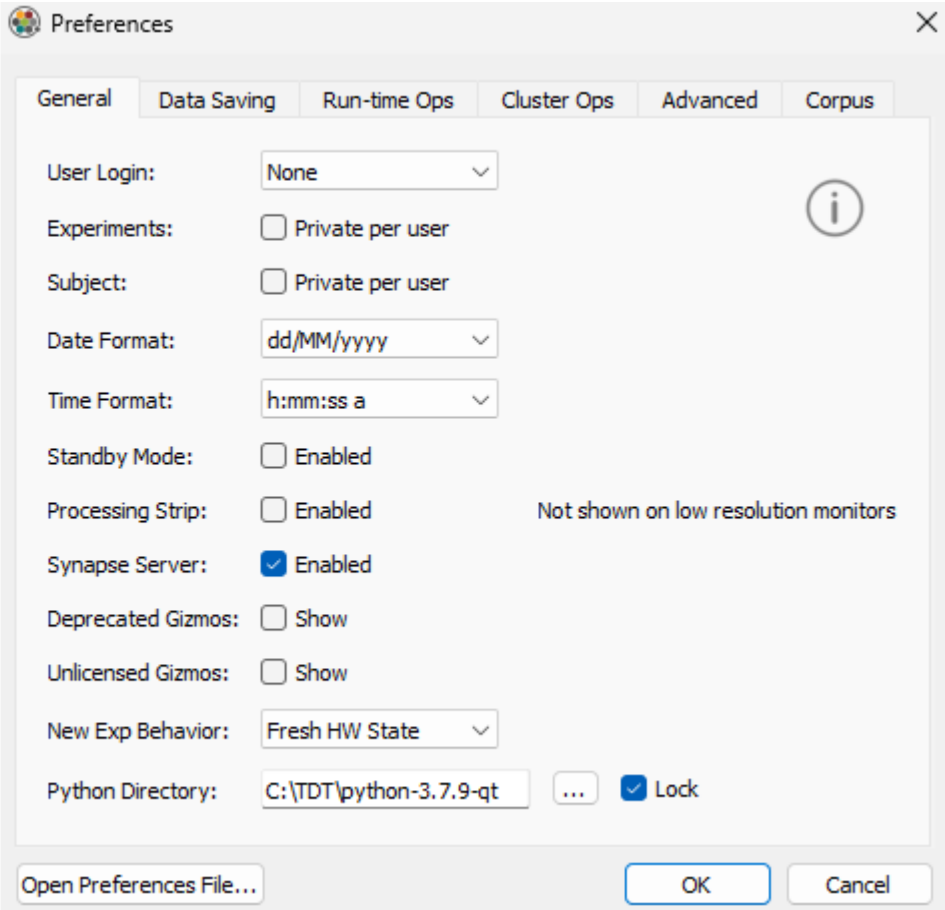
Three blocks might be:
MouseC5-151008-105134
MouseC6-151008-110526
MouseC7-151008-112049

> **Important**
>
> The maximum block name length is 31 characters.

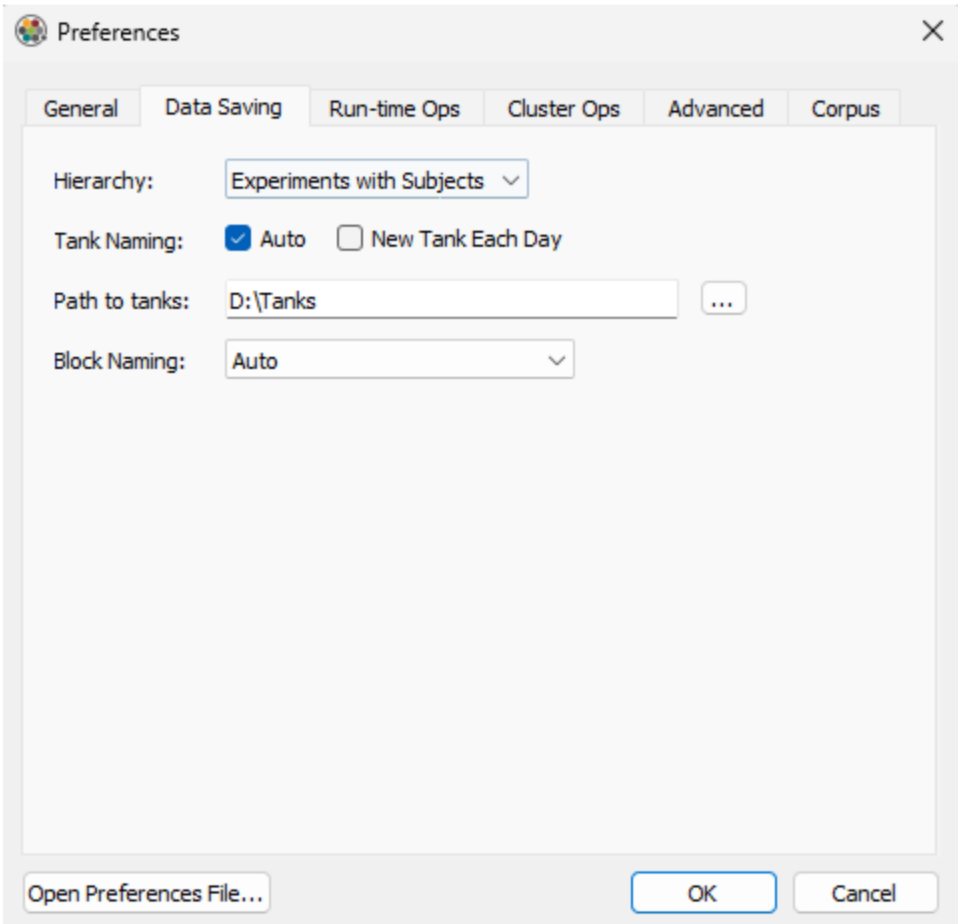
These default preferences are ideal for labs that run an experiment on multiple subjects then move on to another experiment. If your lab does things differently, such as running experiments on the same subject to compare results over time, you can change the hierarchy preference to **Subjects with Experiments** . With this hierarchy, tank names are generated using the subject name and blocks are named using the experiment name. Synapse preferences also allow you to choose a tank path, a different time and date format, when to generate a new tank, or to name tanks manually.

To view the Synapse Preference dialog:

- Click **Menu** at the top of the bar and then click **Preferences**.



Set Date and Time Format in the General Tab



Set Tank Naming in the Data Saving Tab

Accessing Stored Data

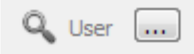
Synapse uses TDT's DataTank format. Data can be can be viewed using OpenScope, OpenExplorer, and OpenSorter, or accessed for analysis via MATLAB. The location for data storage can be set in the Preferences dialog. Data can also be accessed using the History window.

History Window

The History window displays a filtered list of all recording sessions and displays a timestamped list of all changes made to experiments during each session. It is used primarily to access the experiment configurations settings and changes, but it also provides a number of quick data access tools.

To open the History window:

- Click the **User Search/Browse** button, or go to Menu → History.

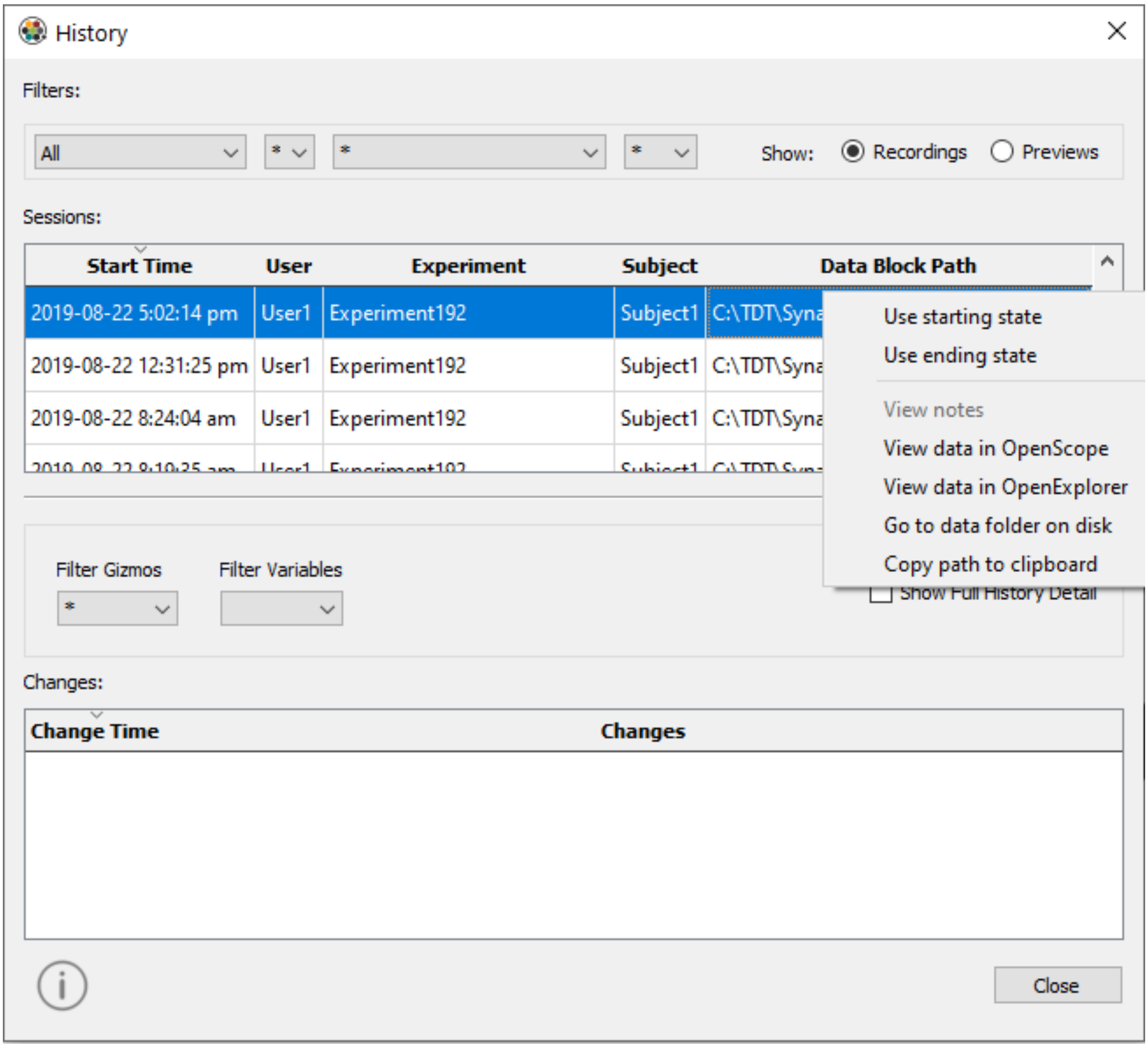


Sessions

The sessions area is the top section of the window. Each row in the upper section contains information about the data collected for a single recording session, including the data path. Several commands for accessing data are available on the shortcut menu.

From the shortcut menu (right-click a row), you can:

- View any notes you made during that recording.
- View the selected data in OpenScope.
- View the selected data in OpenExplorer.
- Open the folder containing the selected data Tank.
- Copy the path for the selected data to Clipboard, for pasting into MATLAB.



History Window

Managing Users and Subjects

Lab Management

The Synapse relational database is the key to the software's powerful lab management features. Synapse uses it to track and save all aspects of your experiments and every aspect of your interaction with the interface. It contains the who, how, and what of each Synapse session, virtually everything except the acquired data. Managing Data for Your Lab, explains how Synapse uses key categories of information in its relational database, like experiments and subjects, along with Synapse preferences; to name and manage acquired data files. This section explains how Synapse user and subject features help you manage how people in your lab interact with the software.

Users

Lab managers can use Synapse preferences related to users, to create and assign user accounts to lab groups or individuals, decide whether passwords are required at log in, and control how experiments are shared within the group. In the relational database, user names are linked to experiments, subjects, parameter changes, and the windows layout.

By default, user features are not enabled. Some features, such as filtering and the History window, that will be discussed in this section can be used in the default state. However, enabling user login greatly increases their utility and benefit.

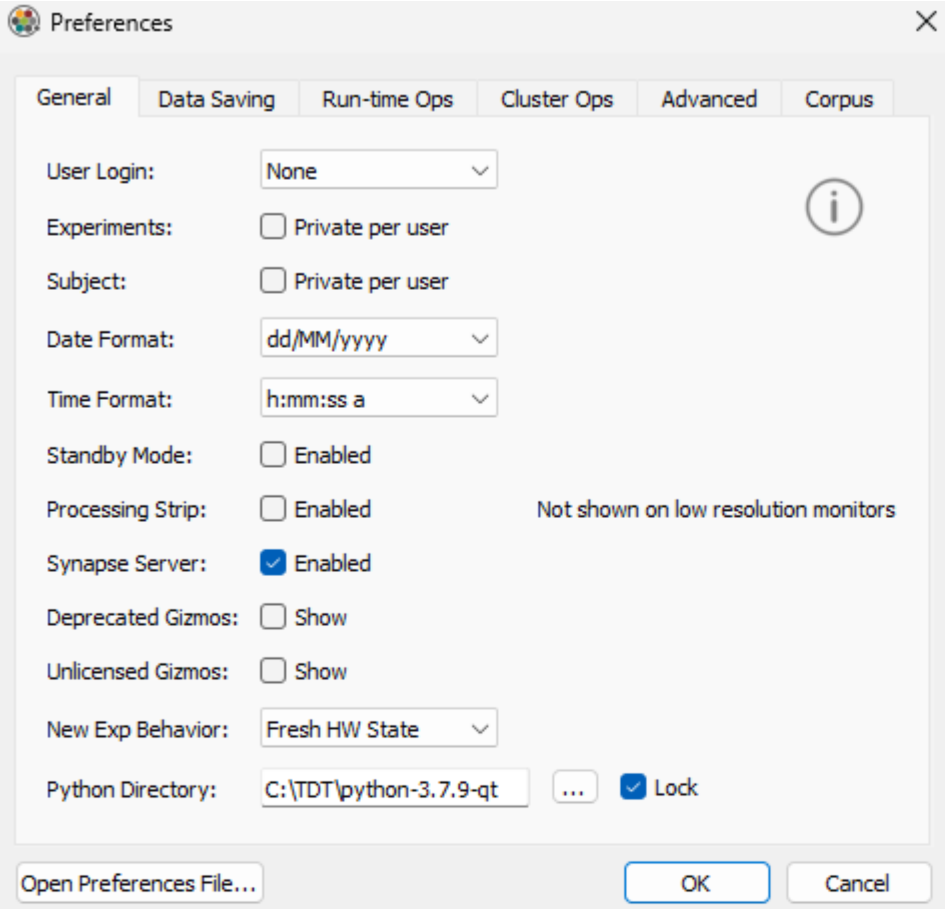
The user functionality allows for individual labs to determine how user names will be used, with the most common being that every person might have there own user name. However, user names can be used to create roles or groups. You'll need to consider how your choice will work with other preferences settings, such as privacy options, before making a final decision.

There are also two different user modes available: with or without passwords. The password functionality is not a security feature. It provides an extra layer of caution to encourage users to login to their own user ID so that logs, change tracking, and filtering will be more effective.

Passwords apply only to experiment configurations, all data are available to all users in the data tank.

To enable User login:

1. Click **Menu** at the top of the command bar and then click **Preferences**



Preferences Dialog

2. For User Login, choose **Required** or **Required with Password**.
3. Optionally, select any of the following options:

   Experiments Check Box - select to make experiments private per user.

   Subject Check Box - select to make subjects private per user.

Privacy

The purpose of privacy in Synapse is to aid in filtering and to minimize mistakes and confusion. When experiments or subjects are private, they are only available to the user that created them. Privacy is controlled in two places.

In the Preferences dialog, the **Experiments Private Per User** and **Subject Per User** check boxes make privacy the default sate for new experiments or subjects created. It doesn't change the privacy of existing experiments or subjects.

Selecting either check box in the Preferences dialog makes privacy the default, by enabling the **Private** check box in the dialog used for creating new users or subjects. However, this check box is available and can be selected or cleared, regardless of the preference setting. This allows users to choose to make particular experiments or subjects private.

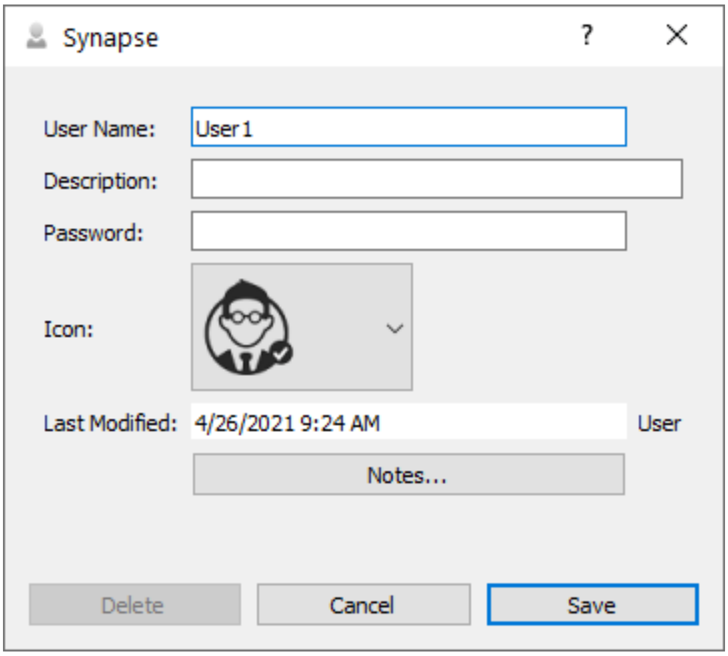
Adding and Selecting Users

When user login is required (in the Preferences dialog). A **User** button is added to the command bar.

Adding Users

To add a new user:

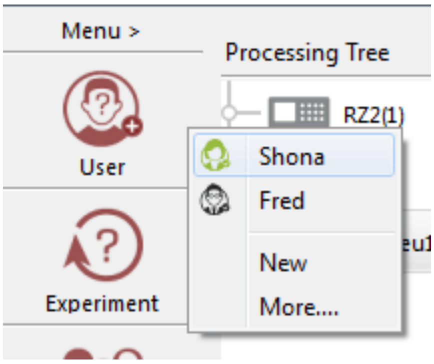
1. Click the **User** button on the command bar, then click **New**.



New User Dialog

2. Enter a name in the **User Name** field. You can also enter a description, password, or notes, and choose a user icon.
3. Click **Save**.

After users have been added, they are available on the shortcut menu or by clicking **More**.



User Category Menu

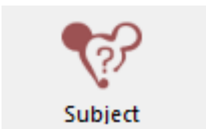
Important

When users are assigned, Synapse is able to store windows layouts for each user and for each experiment. Runtime configuration settings can also be saved by user and experiment and this information becomes part of the 'best' persistence, that is, the best persistence is the most recent runtime settings for the current experiment AND the current user. See Using Persistence with Users and Subjects below.

Subject

Depending on the work done in your lab, you may have just a few chronically implanted subjects or you may have many subjects used for screening. Whatever your lab's work style, the subject name plays a special role in naming data tanks. Synapse easily adapts to either style using the settings in the Preferences dialog and in the category dialogs.

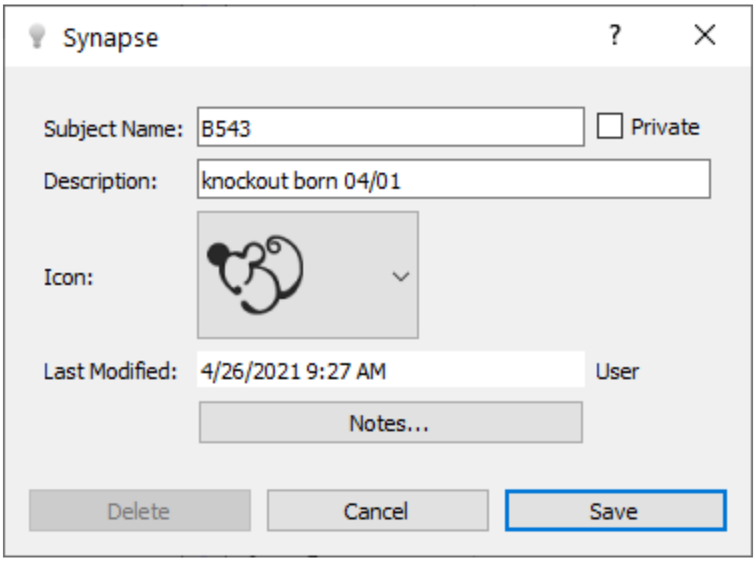
You can select, edit, or add subjects using the **Subjects** category button on the command bar.



Adding A Subject

To add and a Subject:

1. Click the **Subject** button in the command bar, then click **New**.



New Subject Dialog

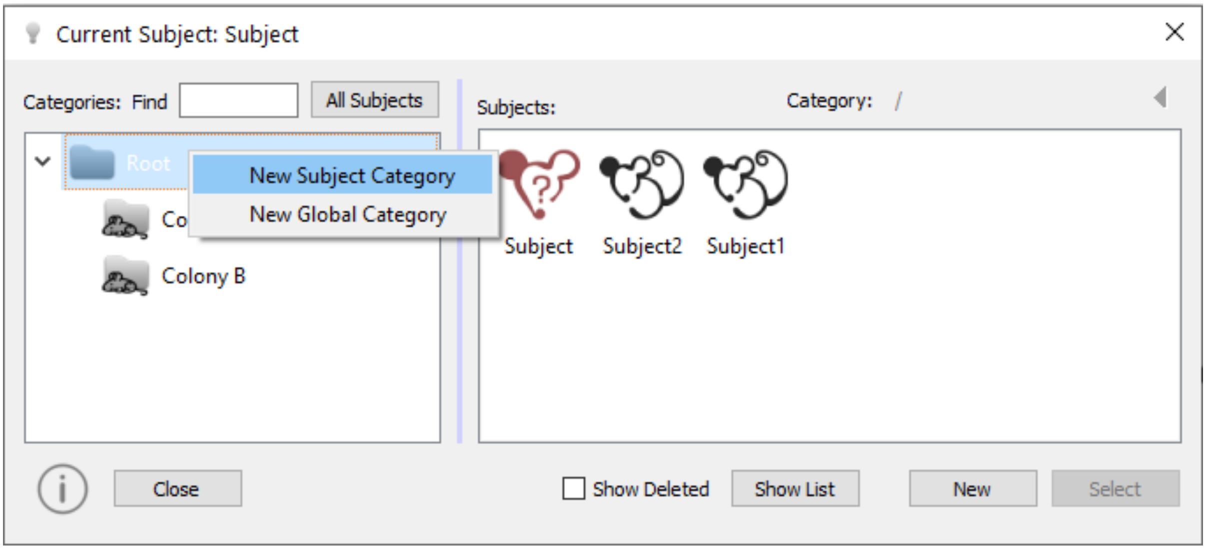
2. Enter a name in the **Subject Name** field. You can also enter a description, password, or notes and choose a user icon.
3. Click **Save** to add the subject.

The new/edit subject dialog also includes a check box to make a subject private. This setting is tied to the **Subject: Private per user** option in the Preferences dialog. It has no meaning unless user names are enabled.

After subjects have been added, they are available on the shortcut menu or by clicking **More**. You can also add multiple subjects from this dialog.

Organizing Subjects

Making subjects private by user is one way to help organize subjects. Another way it to use subject categories in the Subject window. You can find this window by clicking **More** on the Subject shortcut menu. It works similarly to a Windows folder window, with categories (or folders) on the left and subjects (the contents of the folder) on the right. You can create new categories and subcategories by right-clicking the **Root** folder or an existing category folder.



Subjects Window

Using Persistence with Users and Subjects

The Persistence runtime interface was introduced in See Launching Your First Experiment.-See Using the Runtime Interface. on Persistence. It is worth taking another look at how persistence relates to users and subjects.

With a single user and a single subject, persistence ensures that your runtime settings, such as clustering definitions, filter settings, and display options are retained when you switch from preview or record mode to idle.

Fresh

The **Fresh** option is the easiest to explain and understand. It allows you to return to a fresh start by using the experiment's default settings for each gizmo at run time.

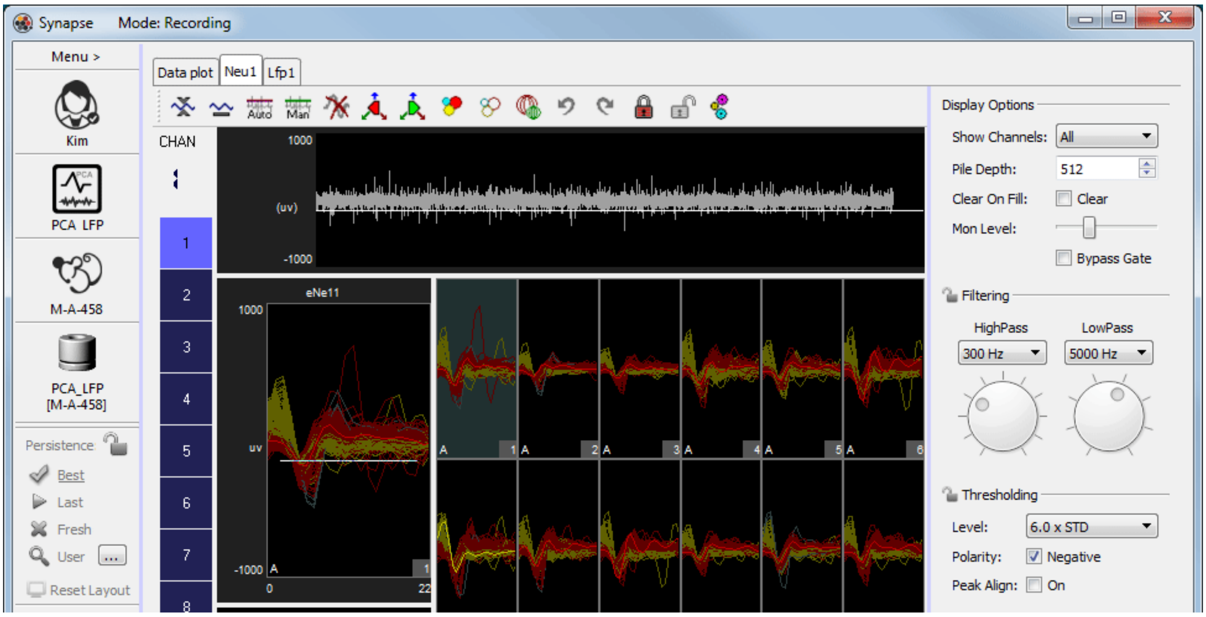
Best and Last

By default, Synapse uses the **Best** option, which applies the most recent settings for the current experiment and current subject.

The **Last** option uses the most recent settings, regardless of subject or experiment. This is useful if you set up your cluster parameters for an animal, but then switch experiments. You can import the settings from the previous run directly into this new experiment.

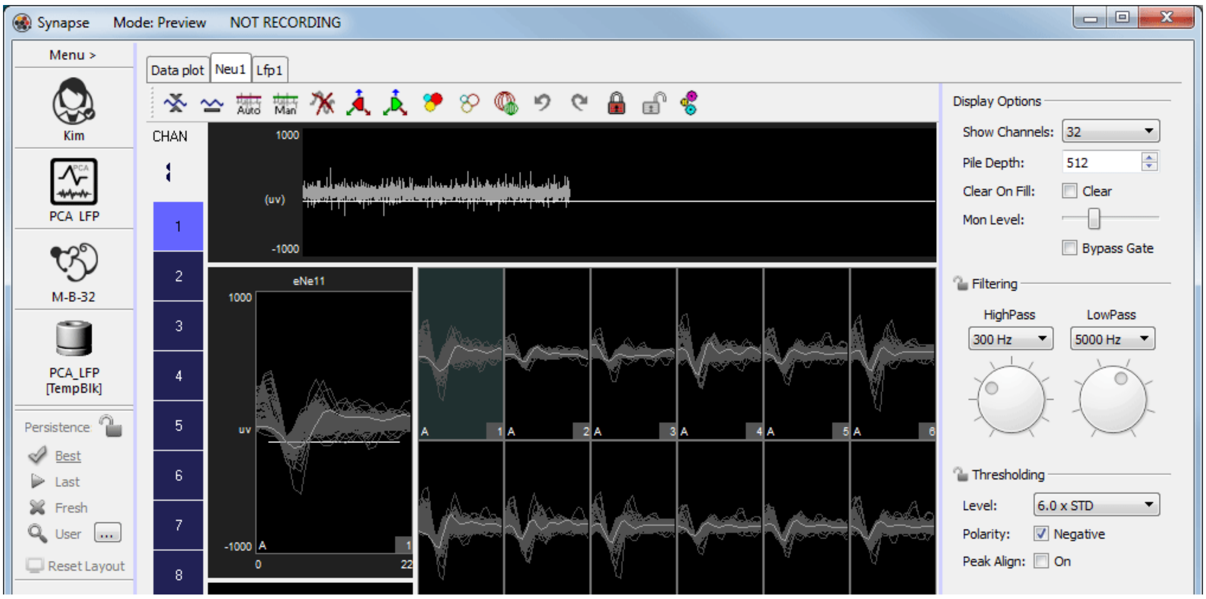
Lock Icon

Even if you switch to another to another type of persistence, such as **Last**, Synapse will return to **Best** when you return to Idle mode unless you lock the Persistence. Initially, the difference between these two options can be difficult to see.



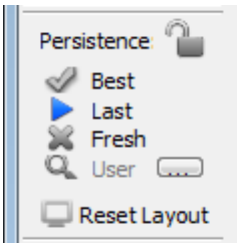
Runtime Window After User Adjustments

Because a different subject is being used, the saved settings are not applied. You see a fresh window.



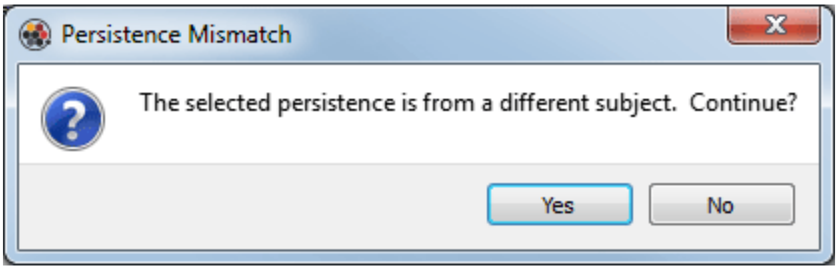
Runtime Window with Best Persistence and New Subject

You may or may not want this behavior depending on the type of experiment. For example, if you are doing a behavioral experiment and using synapse primarily to present stimuli, you might want to apply the last settings regardless of a change of subject. In that case, you would select **Last** persistence.



Persistence Interface - Last Selected

If you choose **Last** and Synapse detects a new subject being used, you will see a warning like the one below. Click **Yes** to continue or **No** to stop and either change the subject or choose **Best** preference.



New Subject Alert - Persistence Mismatch

The more you become comfortable with the persistence features, the more you will realize how helpful they are and how much you already rely on them.

Layout Persistence

It is important to note that the arrangement of windows/tabs, called the 'Layout', is tied to the current User and Experiment. Click the **RT Layout** button at the bottom of the Persistence area on the command bar to change the layout.

You can clear the layout for the next recording, import layouts from other experiments/ recordings, or setup the flow plots for the next recording.

Synapse Fundamentals Reference

Hardware Configuration

The hardware devices that make up your system were carefully selected from System 3's diverse group of signal processors, amplifiers, and input/output devices. Each device has particular features; such as the number of DSPs in an RZ processor, maximum number of recording channels, or types of optical ports available for amplifiers or other peripheral devices. Synapse keeps track of these details for you and offers choices suitable for your system as you're building an experiment. You select the parts of the experiment you want and Synapse generates it, optimized for your hardware.

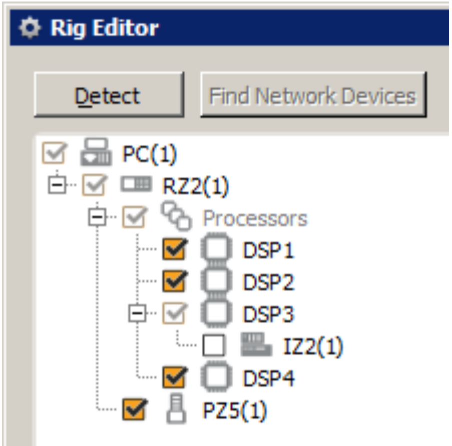
The Rig

Synapse stores information about your hardware in a rig (*.synrig) file. The first time you run the software, you will need to configure the rig. Launching Your First Experiment includes information about how Synapse does some of this for you, detecting principle hardware components and making suggestions for devices that might be present, but can't yet be detected.

Often your rig needs to be configured once then you can forget about it. Your rig configuration selections should match your actual physical system. The rig is not specific to an experiment and limiting the capabilities of your system by disabling I/O channels on a device or disabling a device completely is likely to cause problems in designing future experiments.

How Auto-Detection Works

When you click the **Detect** button in the Rig Editor, your equipment should be displayed in a hierarchical tree diagram.



Hardware Rig

If your devices are not detected, check to ensure your system is properly connected and powered on then retry. If you are currently unable to connect devices, but want to continue to use the software for design or debugging, see Working Without your Hardware.

The Detect feature communicates with the RZ processors to determine which RZ device is connected and to identify the DSPs installed in the device. Common devices which **could** logically be connected are added to the tree in a disabled state, which means that device won't automatically be available when you build a new experiment. You need to review the tree to verify that it correctly represents your system and to enable/disable, add, or configure devices as needed.

PZ5 amplifiers are auto-detected starting in v96. The PZ5M **does not** auto detect and must be added manually.

The DSPM can auto-detect PZ5, SI, and iCon devices. If you run the auto-detect with no device connected to the RZ or powered on, then the DSPM will not correctly auto-detect a device. Power on and connect the peripheral device and reboot the RZ before you try to auto-detect again.

RX devices are identified, but information about the number of DSPs (2 or 5) and analog I/O configuration of the RX8 is not automatically detected and must be set by confirmed by the user.

Understanding the Hardware Tree Diagram (Rig)

The tree diagram in the Rig Editor represents your hardware in a hierarchical way and all branches start or end with the PC at the top of the tree. The icons below the PC represent the devices detected, added by a user, or predicted/suggested by Synapse as a device that might exist based on known information it has about the processor device(s) in the system.

For example: An IZ2 will be added if your RZ2 houses a DSP-I card, but the device is disabled by default.

DSP-I with predicted IZ2 (disabled)

Disabled Devices and the Processing Tree

Disabled devices can later be added to an experiment in the Processing Tree using the **Add HAL** command in the right-click menu of the parent RZ device. Also, if an experiment requires a device that is disabled in the Rig, it will be added to the Processing Tree automatically when an experiment is opened.

Specialized DSPs and Related Devices

DSP labels provide additional information about the type of DSP and its associations. The label takes the form **DSPX#** where:

X = device type to which the DSP connects, such as amplifier or RS4 data streamer (omitted if none) or indicator for multi-core DSP
= logical number (or index) assigned to the DSP within the RZ device

For example:

Fourth DSP in an RZ, specialized DSP-I for IZ2 stimulator

DSP Type/Device Connection Key

M = Subject Interface, iCon Behavioral Control Interface, PZ amplifier

S = RS4 data streamer

V = RV2 video processor

I = IZ2 electrical stimulator

U = PO8e streamer

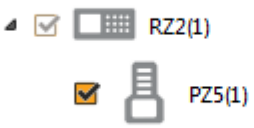
P = PZ amplifier

Q = Quad Core processor

> **Note**
>
> The optical version of the DSPQ card can also support any of the device connections (such as video processor or amplifier). This is not indicated in the DSP name.

Amplifier Connections

If Synapse detects an RZ2, a PZ5 icon is added below the RZ2 icon. This is because the RZ2 has an amplifier port and a PZ5 is typically used with this device, but Synapse can't detect the amplifier model (PZ5, PZ2, or PZ3).



RZ2 with Predicted PZ5

If you are using a different amp or a different number of channels than the default, you will need to change the configurations options as described below.

Each device can be individually configured using its device options. When a device is selected in the rig tree, any configurable features are displayed to the right. Options may include model number, channel count, and so forth.

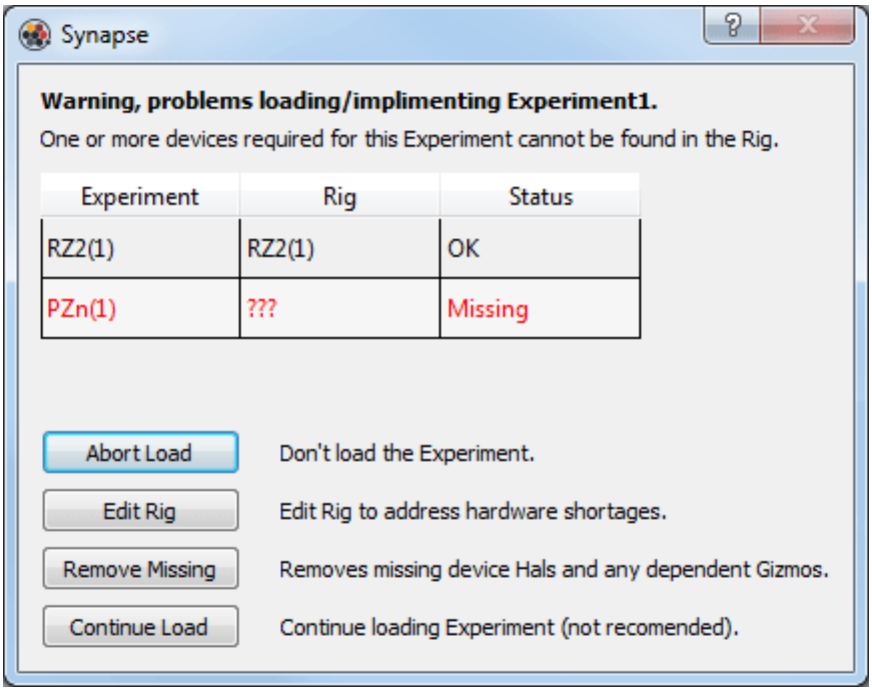
Add/Remove Devices

Use the right-click menu in the rig tree to add any devices in your rig that didn't automatically configure.

To prevent loss of existing experiment hardware configuration, keep the **Merge Previously Saved Configuration** check box selected. To refresh all of the hardware objects in the Processing Tree to their default state, uncheck this box.

Experiment - Rig Mismatch

When you open an experiment, Synapse checks the Rig to determine if the devices required for the experiment are available in the processing tree. If a mismatch is detected, a warning window is displayed. The top of the window displays the devices required by the experiment and their status in a grid.



Rig - Experiment Warning Window

The lower half includes a variety of options for handling any problem.

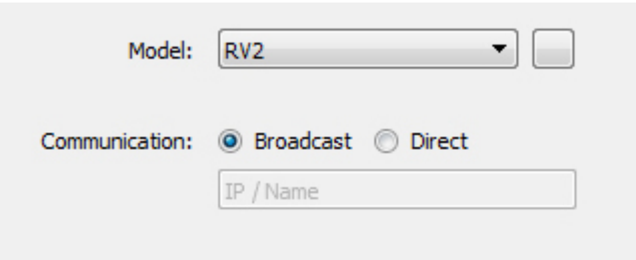
Rig Specific Device Options

Network Devices

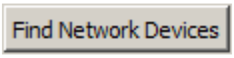
Network enabled devices, that is the RS4 and RV2, must be configured for network communications in the rig.

To configure a network device:

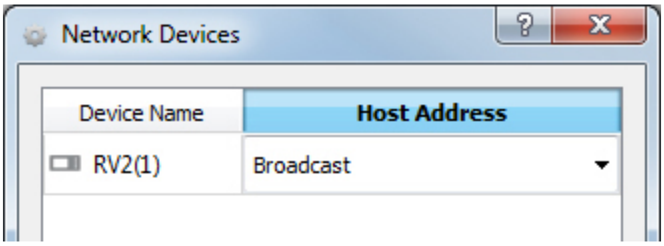
1. Make sure the devices is enabled in the hardware tree, then click to select it.
2. In the Options area, select the Broadcast or Direct connection radio button, then enter the IP address.



3. Alternatively, click the **Find Network Devices** button.



4. In the Network Dialog bog, select the IP address in the Host Address drop-down for the selected device.



Working Without your Hardware

If you have installed Synapse to a computer that is not connected to a hardware system, you can build a phantom rig for planning and debugging. You can also spec out a system that would run any experiment you designed.

1. In the Rig Editor, right-click the computer icon and click **Add RZn** on the shortcut menu.

   If necessary, select the RZn check box.

In the area to the right, the default Model and I/O settings for the selected hardware is displayed.

2. If the model shown doesn't match, click the **Model** drop-down menu and click your model in the list.
3. Repeat this process to add DSPs, amplifiers, and any additional device to the diagram.
4. When all the necessary hardware has been added, click the **OK** button.

In the Synapse designtime window, the Processing Tree is populated with the hardware in your rig. With your phantom rig you can configure your experiment.

Close Synapse and launch the Corpus Hardware Emulator. Corpus will emulate the rig you just defined. When you launch Synapse while Corpus is running, Synapse will connect to Corpus and use the emulated rig so you can test your experiment using CPU resources. See Remote Experiment Design for more information.

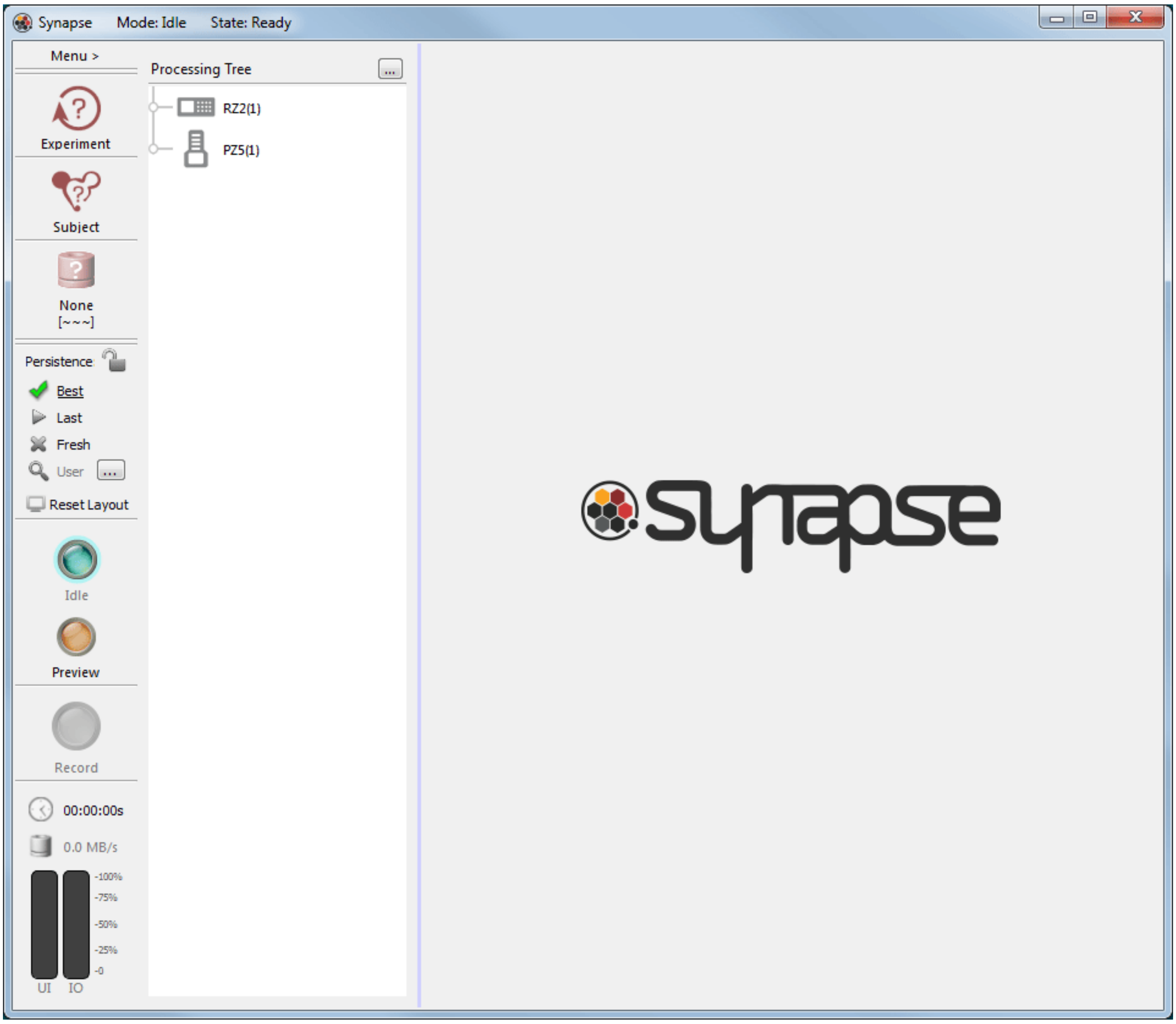
Make sure your actual hardware system matches your rig configuration before running your experiment on real hardware.

Import or Export Rigs

The Rig Editor includes **Import** and an **Export** buttons. These buttons can be used to open an existing rig file (Import) or preserve the current rig for future use (export). These buttons launch Select a Hardware Rig or Save the Hardware Rig dialogs that function much like a typical Open or Save as dialog.

Designtime Reference

When you launch Synapse, you see a streamlined user-interface that automates all but the highest level set-up tasks for you. This is the designtime interface where you can make choices about things like what type of data to collect and what threshold, sorting, or other processing tasks to include in your experiment.



The Synapse Designtime Window

The window is divided into three areas:

1. The **Command Bar** contains the most often used elements of Synapse.
2. The **Processing Tree** displays a graphical representation of your experiment.

3. The **Details Area** displays setting and configuration options at designtime or plotting and control windows at runtime.

At each step of the design process, Synapse uses information about your hardware and selections you've already made to show only the relevant choices. Once added, you can review and modify the settings in the details area. Often, you won't need to make any changes at all. While Synapse supports drilling down to every detail of how the system works, it has also been designed to make that unnecessary for most Synapse users.

The Processing Tree

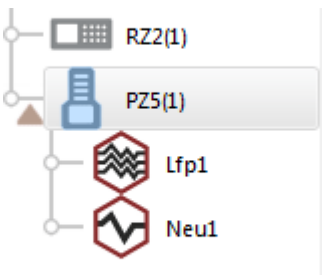
The Processing Tree is both a graphical representation of the processing tasks that make up your experiment and a design tool. The tasks added to the tree along with how they are ordered and connected forms the processing instructions that will be loaded to the hardware at runtime. Each tree can have many parts or branches.

Devices with input/output functionality form the trunk of the tree. They represent the starting point (or potential starting point) for a branch.
RZ devices appear in the Processing Tree to represent their front panel analog or digital inputs.
Acquired biological signals are often input by a PZ amplifier, so it also appears on the trunk of the tree.

> **Important**
>
> If an input/output device does not show up automatically in the Processing Tree, you will need to edit the rig. See The Rig.

Tasks, such as filtering, signal processing, and data storage form the branches of the tree diagram. They are added to the device that will be used to input or output the signals associated with a particular task.

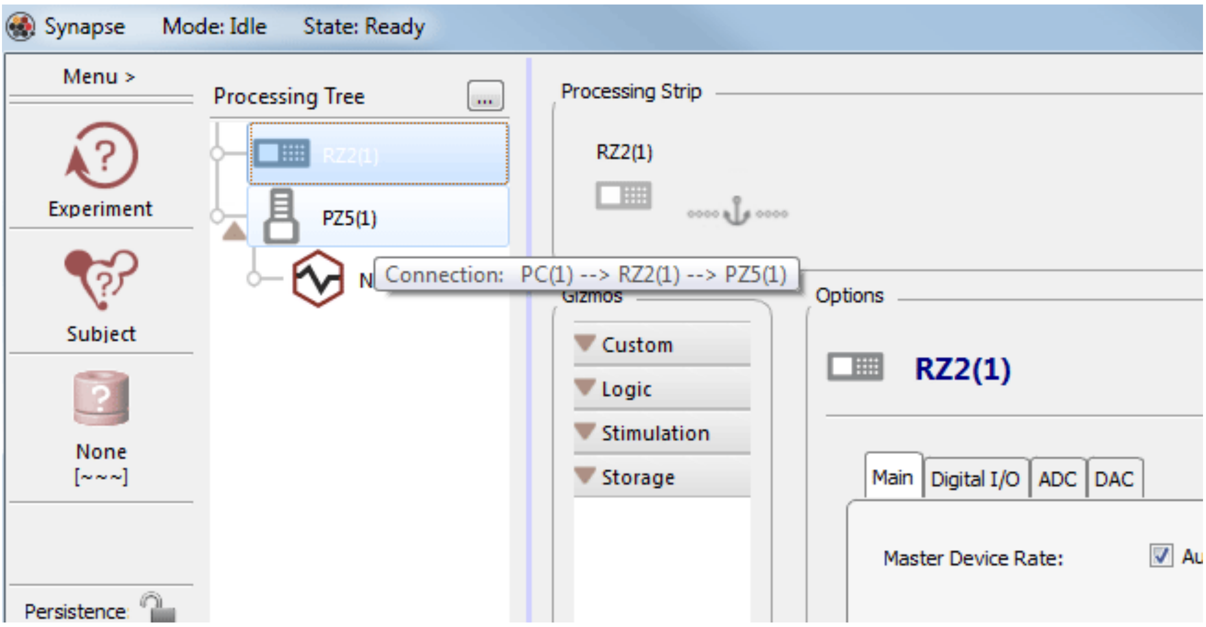
 Click the triangle/arrow to expand or collapse a branch.



Biological signals are typically acquired on a PZ amplifier; so neural processes, like spike detection, are added to the PZ in the tree.

Multiple tasks can be added to any branch, in parallel or one after another with tasks ordered logically.

Mouse over a device to see how the hardware devices are connected.



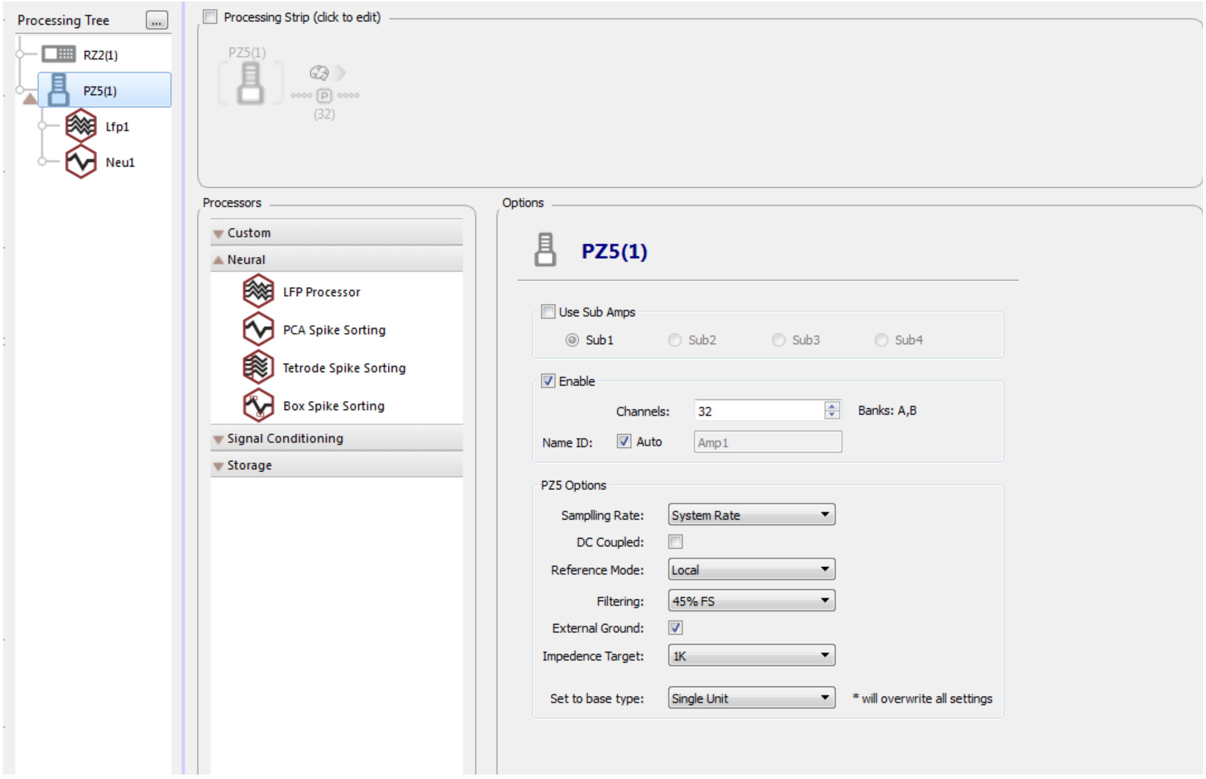
Context Sensitive Hardware Connection Tip

Using the Processing Tree

The Processing Tree is a simplified view of the experiment. The specifics can be more closely examined in the details area. When an item is selected in the tree, the details area is divided into three sub-areas: the Processing Strip, the Gizmos list, and the Options area.

The information displayed in these three subareas is specific to the selected item or branch:

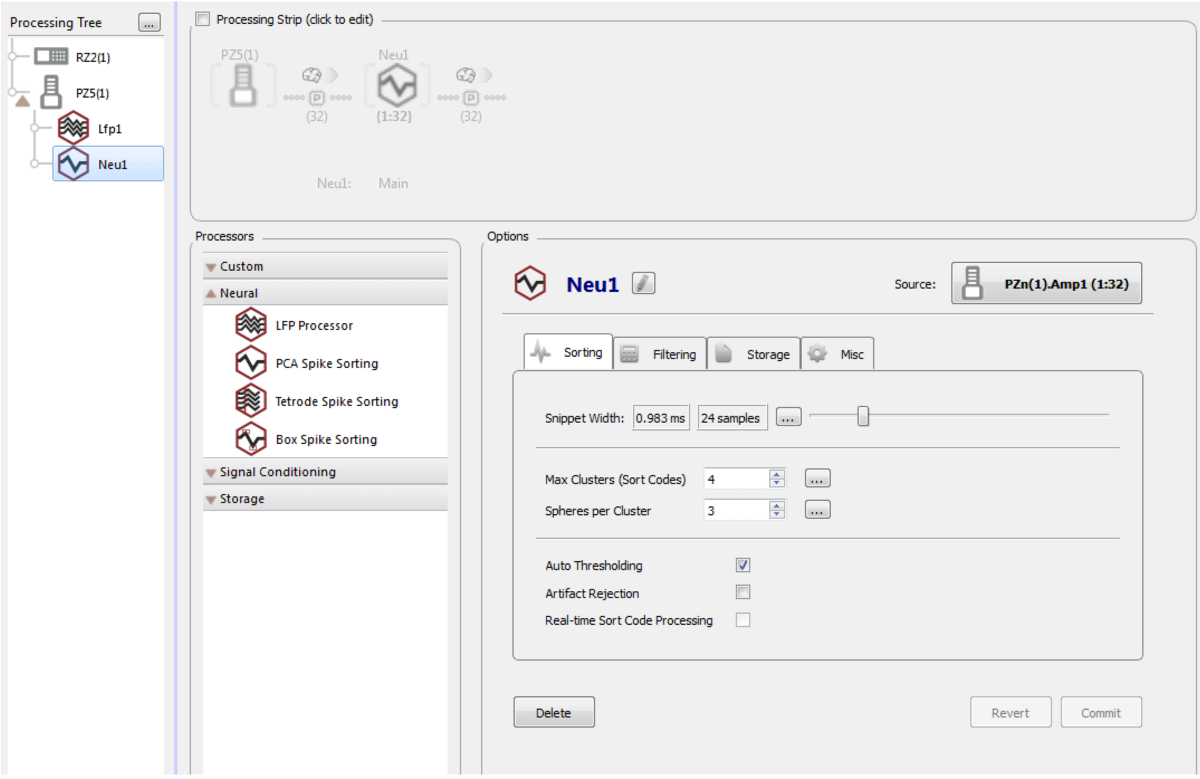
1. The **Processing Strip** - displays a drilled down look at the corresponding branch of a signal or data path, following the first inputs of each gizmo in the path.
2. The **Gizmos** list - displays tasks that can be added to the selected item
3. The **Options** area - displays configuration options for the selected item



Synapse Designtime Window

Selecting a hardware device in the Processing Tree updates the Gizmos list to show only the tasks appropriate for that device and displays configuration information for the device in the Options area. This information is saved as part of the experiment and is also referred to as a HAL (Hardware Abstraction Layer) because it gives Synapse everything it needs to manage the hardware- related low level programming tasks.

Similarly, selecting a gizmo in the tree displays the settings for the gizmo, typically arranged on tabs in the Options area.



Designtime Window with the Neu1 Gizmo Options Displayed

To reset a device or gizmo:

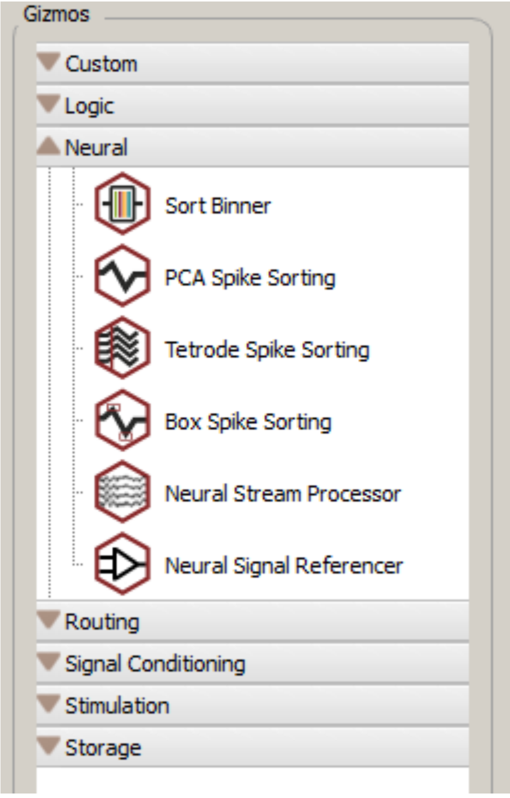
- Right-click the corresponding icon in the Processing Tree and click **Reset to Default** on the shortcut menu.

To add a previously unused HAL:

- Right-click a related icon in the Processing Tree and click **Add HAL** on the shortcut menu.

  This option allows you to add a HAL for a device that is present in your Rig, but previously not in use in the experiment. By adding the HAL directly in the Processing Tree, you avoid returning to the Rig Editor which resets all of your device HALs.

The Gizmos List



Only gizmos that can connect to the currently selected object in the Processing Tree are shown. Select a different item in the Processing Tree and different gizmos will be displayed.

The gizmos are grouped by task type, such as storage or signal conditioning. For the full list of categories and descriptions, see the Gizmo Categories Reference.

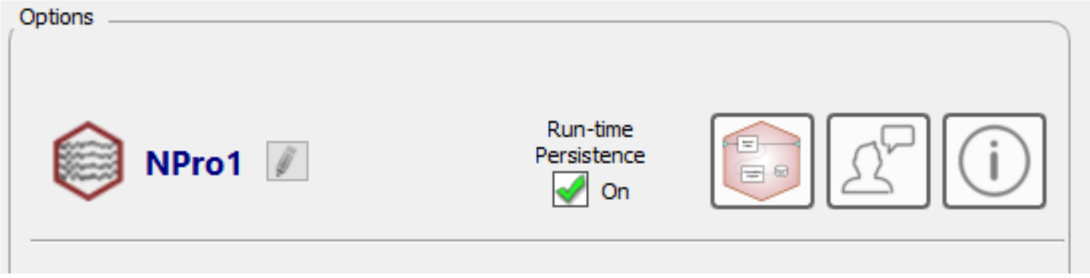
Double-click a gizmo to add it to the selected item in the Processing Tree.

Gizmos in the list are like menu choices, they can be added to the Processing Tree more than once and for more then one input/output source.

The Options Area

Name/Source/Block Diagram

The top section of the options area differs slightly depending the device or gizmo selected. It typically displays an editable device or gizmo icon and name. It may also display the Primary Source for the device or gizmo and the **Block Diagram** toggle button.



The Gizmo Name


The gizmo name is generated automatically for you and is based on the type of gizmo selected. If needed, click the **Edit** icon to modify the name. The name must be at least three letters long and the first three letters are used to form the Storage ID for any related data stores in the data tank. The field turns red if the minimum three letters are not included.

> **Important**
>
> Changing the name of a gizmo that includes data storage will change the name(s) of the storage ID(s) in the data tank. Synapse will display a warning dialog box the first time you attempt to do this. For sanity, changing the store name after you have started collecting data with the experiment is not recommended.

Run-time Persistence

When Run-time Persistence is turned off for a gizmo, the persistence (for that gizmo only) is always 'Fresh', meaning the designtime settings are always loaded when the recording begins. It overrules whatever the global persistence is set to.

If this box is checked, the gizmo follows the global persistence rules.

The Block Diagram Button

The **Block Diagram** button provides access to source settings and the block diagram. See the Gizmo Reference for more information.

The Feedback Button

Send TDT feedback on Synapse, this gizmo, or anything else you want to let us know about.

The Help Button

Opens the Synapse Manual to the current gizmo to learn more about it. By default it opens the online manual in a web browser. If internet is unavailable it opens the local PDF in `C:\TDT\TDTHelp`.

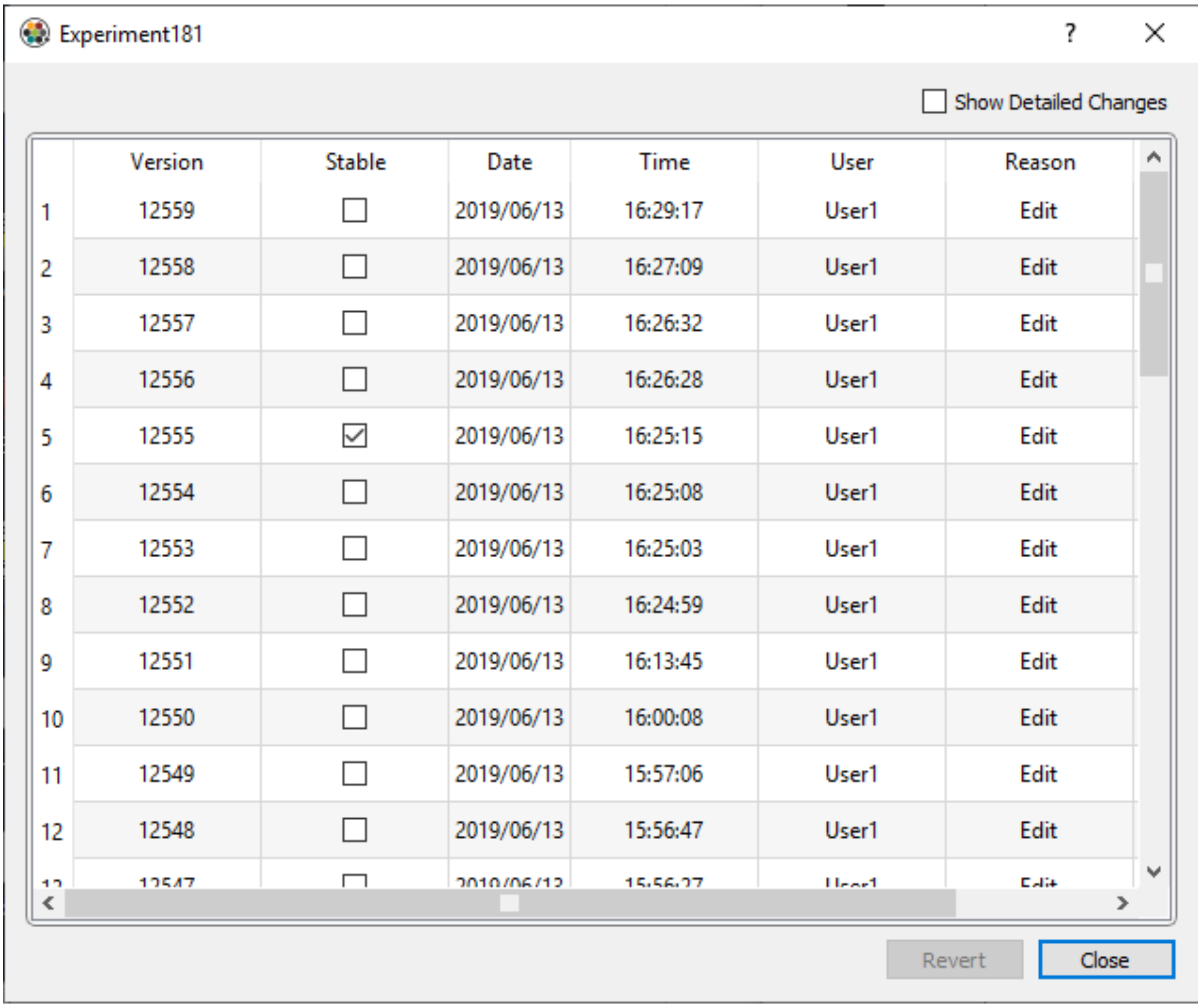
Global Options Buttons

Located below the options tabs and applied across all tabs.

1. **Delete** button - Delete the gizmo from the Processing Tree.
2. **Revert** button - Return to the last saved or 'committed' state. You can also use Shift + F7 keyboard shortcut to revert.
3. **Commit** button - Save changes on all tabs. You can also use F7 on the keyboard to force a Commit.

Experiment Changes and the Revision Log

In Synapse, the experiment is saved each time you make or "Commit" a change. If you need to roll back a change in the experiment design, you can do so in the Experiments Revision Log.



Experiment Revisions Log

You can sort the information by clicking a column header. When you find the version you need, select it and click the **Revert** button. Use the **Show Detailed Changes** checkbox to add a details column with a description of the changes that were made in that revision.

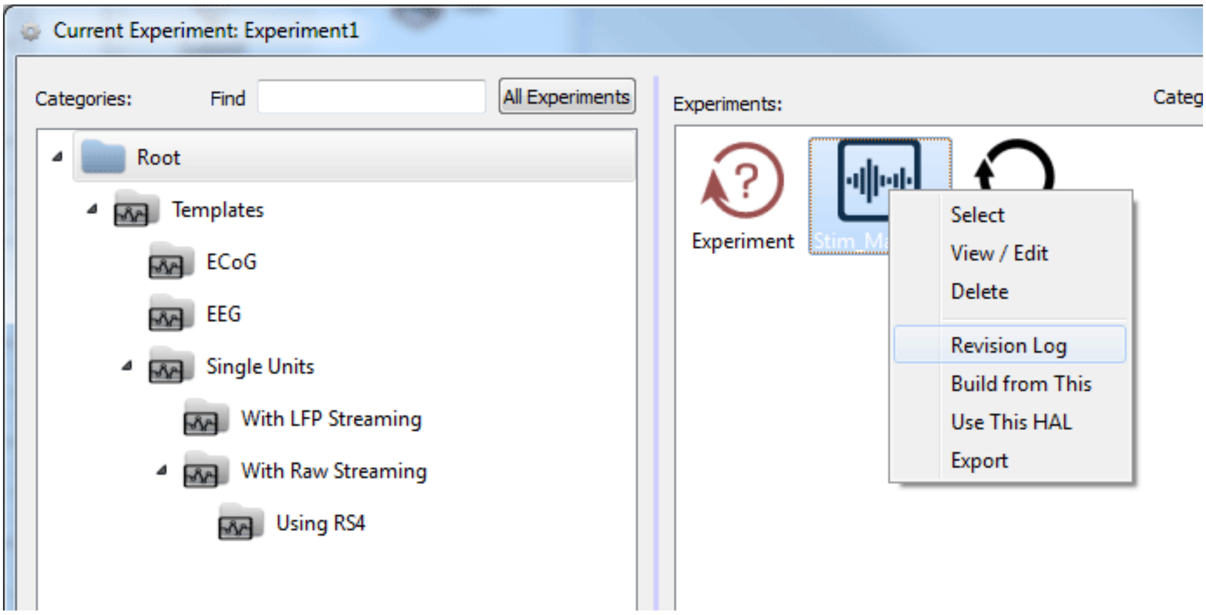
If you are continually making changes but want to revert back to a specific version, check the **Stable** checkbox to flag a particular version you often revert back to. This is a visual indicator for your record only.

Opening the Revision Log

If you have the experiment open and it has a revision history, a "Revision Log" command is added to the Experiment Button menu.

If the experiment is not open:

1. Click the Experiment Button and **More**.
2. In the Experiment Selection Window, right-click the experiment icon and click **Revision Log**.



Experiment Selection Window with Experiment Shortcut Menu Shown

The Connections Diagram

The Connections diagram provides a top-level view of experiment design. The diagram is available from the **Menu** or the **Connections** button, and can be printed.

Each node shows available inputs and outputs. The color of arrowheads on the connection arrows indicates the data type (such as logic or floating point). The arrows are also labeled with the channel range for any multi-channel signals.

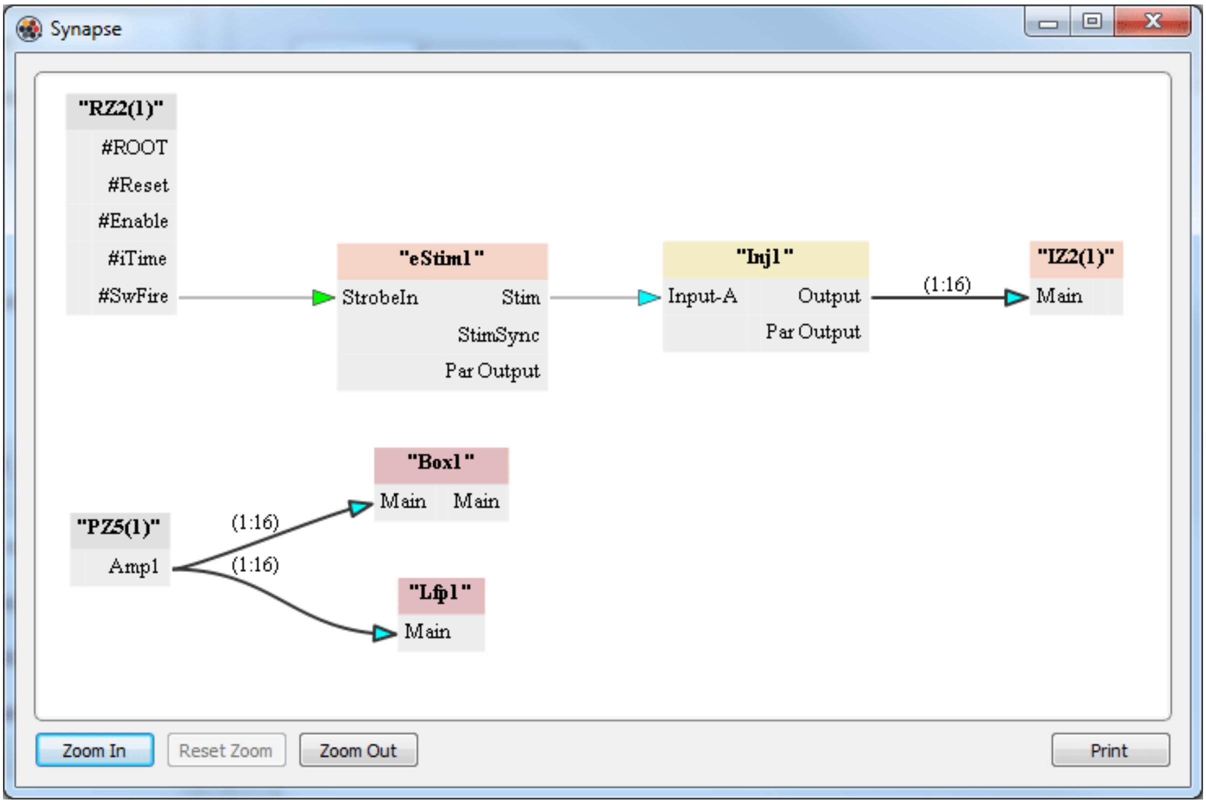
Common data types include:

► floating point type

► integer type

► logical type

The diagram also functions as a debugging tool. You can double-click a node to jump to the selected gizmo or HAL and make changes.



Connections Diagram

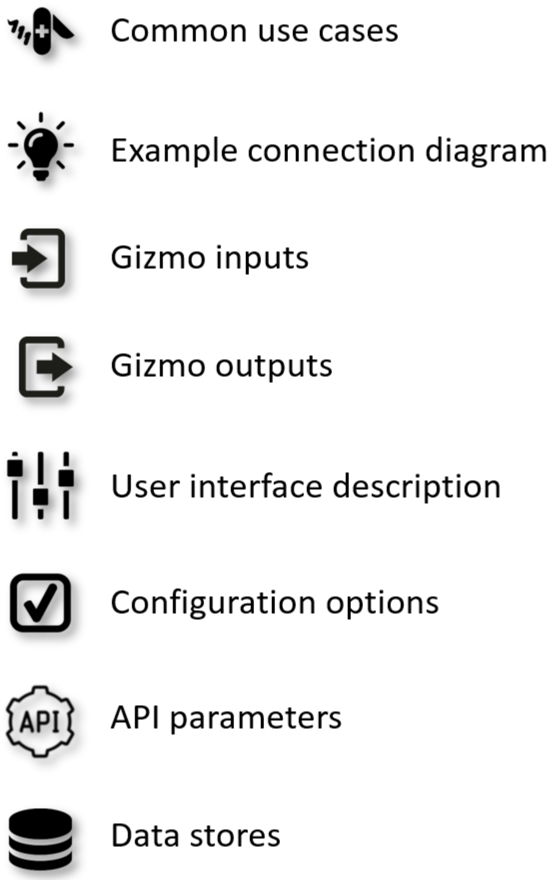
In the diagram above, a logic signal from the RZ2 triggers electrical stimulation. A single channel, floating point signal is passed from the stimulation gizmo to an injector gizmo where it becomes a 16 channel data stream routed to the IZ2 HAL. The diagram also shows data acquired from a PZ5 is passed to two different gizmos for parallel processing.

The Gizmo Slides Window

The Gizmo Slides Window provides a top-level view of the currently selected gizmo. The slides window is available by clicking the '?' button above the Processing Tree.

Each gizmo has a set of slides that show common use cases, example connection diagrams, common inputs and outputs, an overview of the user interface, and more.

When you select a new gizmo, the slides for that gizmo are shown. This is helpful when you are getting to know the user interface and what all the gizmos do.



Gizmo Slide Legend

The Processing Strip

The Processing Strip can be shown as part of the Synapse main window using the Preferences dialog (see Synapse Preferences. It displays a branch of the signal or data path, from the signal source to the item selected in the Processing Tree. The primary purpose of the Processing Strip is to provide information about the signal path at a glance.

The Processing Strip

The number of channels in the signal stream are displayed beneath the gray line representing the signal path. The number shown is the number of channels in the signal at the output of the device or gizmo to the left. In the illustration above, the number of channels at the output of the Map1 gizmo is 16. The number of signals fed into the process was 32, so the illustration shows that the number of channels mapped and passed through does not include all channels.

The numbers below a gizmo display the first channel number in the signal and the number of channels in the signal, that is {first channel:number of channels}. Many gizmos allow the user to change the channel range. This makes it possible to split up a multi-channel signal into several different branches so they may be processed differently.

Menu and Command Bar

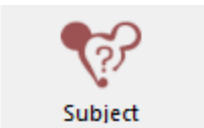
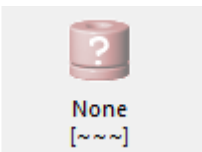
Main Menu

Click **Menu** to display.

Command	Description
Preferences	Launch the Preferences dialog
Edit Rig	Auto-detect hardware or configure devices manually
Clear Session	Clear all experiment settings and return to the default state
Log Window	Open the Log window
Connections	Open the Connections dialog
Notes	See the Notes for the currently selected User, Subject, or Experiment
History	Open the History dialog of past recordings
Clean Storage	Clear database records from old Preview recordings or Empty Data Tank folders that have no blocks in them
Help	Launch the software manual (PDF)
About	View version number and copyright, and update Synapse
Licensing	Activate / de-activate gizmo packages
Exit	Close the program; will prompt to save open experiments
Window>Flow Plot Setup	Set up the flow plot
Window>Merge All	Reset all tabs at run-time to default layout
Window>Show/Hide	Show/Hide specific tabs at run-time

Categories

Click corresponding button to display. Menu options are available in IDLE mode only, unless otherwise specified.

Category	Option	Description
	User Names	Select a user name from the list
	New	Launch a User dialog box to add a new user profile
	Logs	Open a memo dialog box
	More....	Launch the User window where you can choose an existing user or launch the User dialog box to create a new one
	Experiment Names	Select an experiment from the list
	Undo	Return to the state before the last change
	Redo	Redo last action that was undone
	Locked	Locks the current experiment so that no accidental changes can be made to it
	Save As	Launch the Experiment dialog box, where you save the experiment with a name, description, and icon
	Import Experiment	Select a *.synexpz experiment file to import and load
	Export Experiment	Export the experiment along with last persistence and supporting files, such as parameter, stimulation, and map files
	Logs	Open a memo dialog box
	New	Launch the Experiment dialog, where you can create a new experiment
	Revision Log	Open an experiment specific history of changes that can also be used to return to an earlier version of the experiment
	More....	Launch the Experiment window where you can choose an existing experiment or launch the Experiment dialog box to create a new one
	Subject Names	Select a subject name from the list
	New	Launch the Subject dialog, where you can create a new subject
	Logs	Open a memo dialog box
	More....	Launch the Subject window where you can choose an existing subject or launch the Subject dialog box to create a new subject
		Displays current tank and block names. Experiment and subject are used to generate the names according to hierarchy set in the Synapse Preferences. Main label is the string appended to the tank name, sub label is the string appended to the block name

Templates

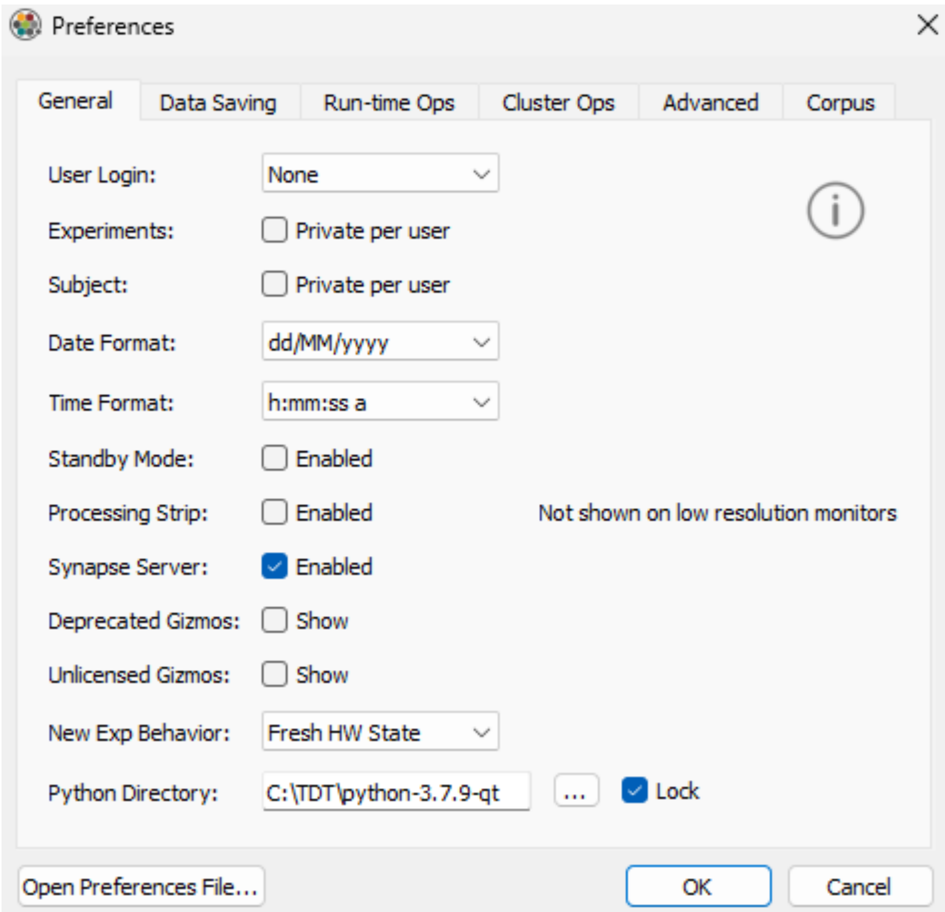
Templates are experiment files that have been created by TDT. Each template is a basic working experiment that can be run as configured, but you will more likely begin with the template and modify it to meet your needs. Templates have been designed to work with rigs suitable for the type of experiment. When opened, Synapse will attempt to adapt the configuration to your hardware rig and alert you to any conflicts it is unable to resolve automatically.

Synapse Preferences

To view the Synapse Preference dialog:

- Click **Menu** at the top of the bar and then click **Preferences**.

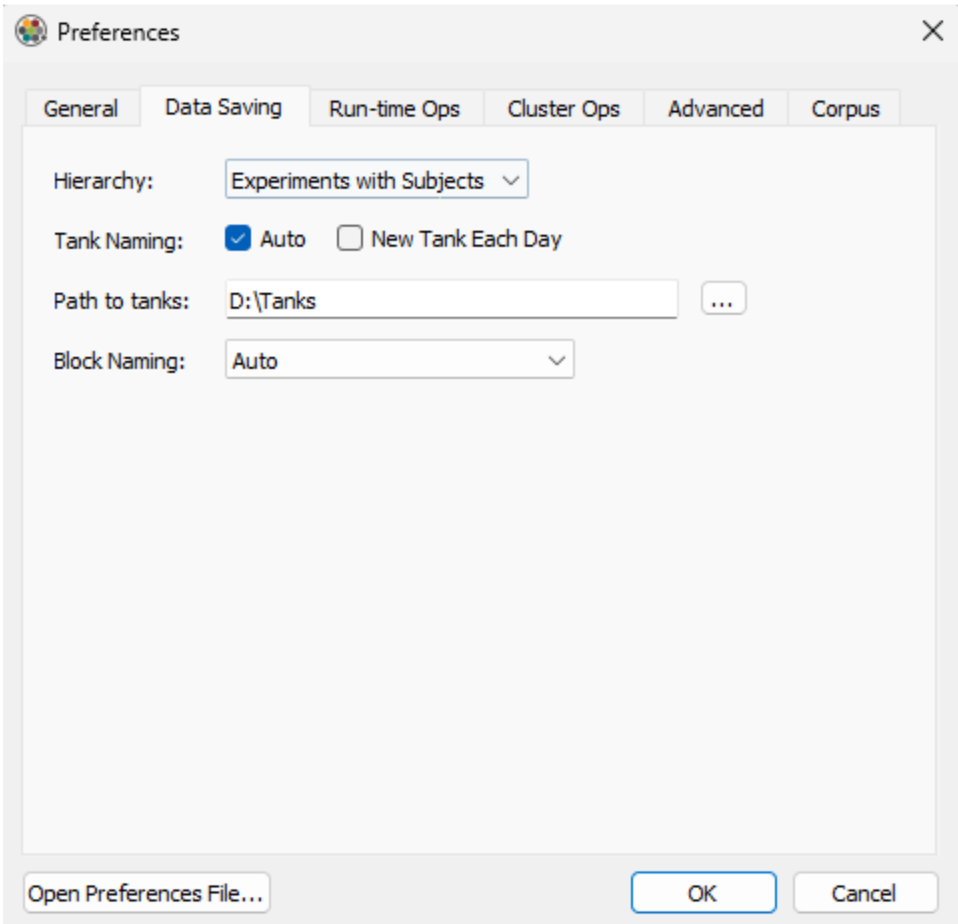
General



Preferences Dialog, General Tab

Option	Description
User Login	Choose a user tracking method. For Required, a User must be selected to start a recording. For **Required with Password**, each user must log in with a password. This is only for organizational purposes and is not a secure method of creating privacy.
Experiments	Select check box to make experiments private per user
Subject	Select check box to make subjects private per user
Date Format	Choose a date format: `MM/dd/yyyy`, `dd/MM/yyyy`, or `yyyy/MM/dd`
Time Format	Choose a time format: `h:mm:ss a` or `hh:mm:ss`
Standby Mode	Enable Standby option at runtime. This mode loads the circuits and starts them but does not send the global trigger to begin acquisition. This mode is useful if loading memory buffers through SynapseAPI; it gives the system enough time to load the buffers before starting the recording.
Processing Strip	Show in the main Synapse window
Synapse Server	Add Synapse Server button in gizmo options so that SynapseAPI parameters and syntax can be displayed. Requires a Synapse restart. See the SynapseAPI Manual for more information.
Deprecated Gizmos	LFP Processor gizmo has been replaced with Neural Stream Processor. The Electrical Stimulation gizmo had a major upgrade to the Electrical Stim Driver gizmo. Check this box to show the old gizmos in the gizmo list.
Unlicensed Gizmos	After the trial period expires, gizmos that aren't licensed won't be shown in the gizmo list. Check this box to show the unlicensed gizmos in the gizmo list. See Licensing for more information.
New Exp Behavior	When creating a new experiment, choose whether to keep the existing hardware settings (for example, the enabled IO on the RZ processor, or the LUX configuration on the RZ10 processor), use a fresh state, or Prompt the user for this choice.
Python Directory	When using the Pynapse gizmo, this option stores the default location of your Python installation and the default Python environment to use for Pynapse. See Pynapse Manual for more information.

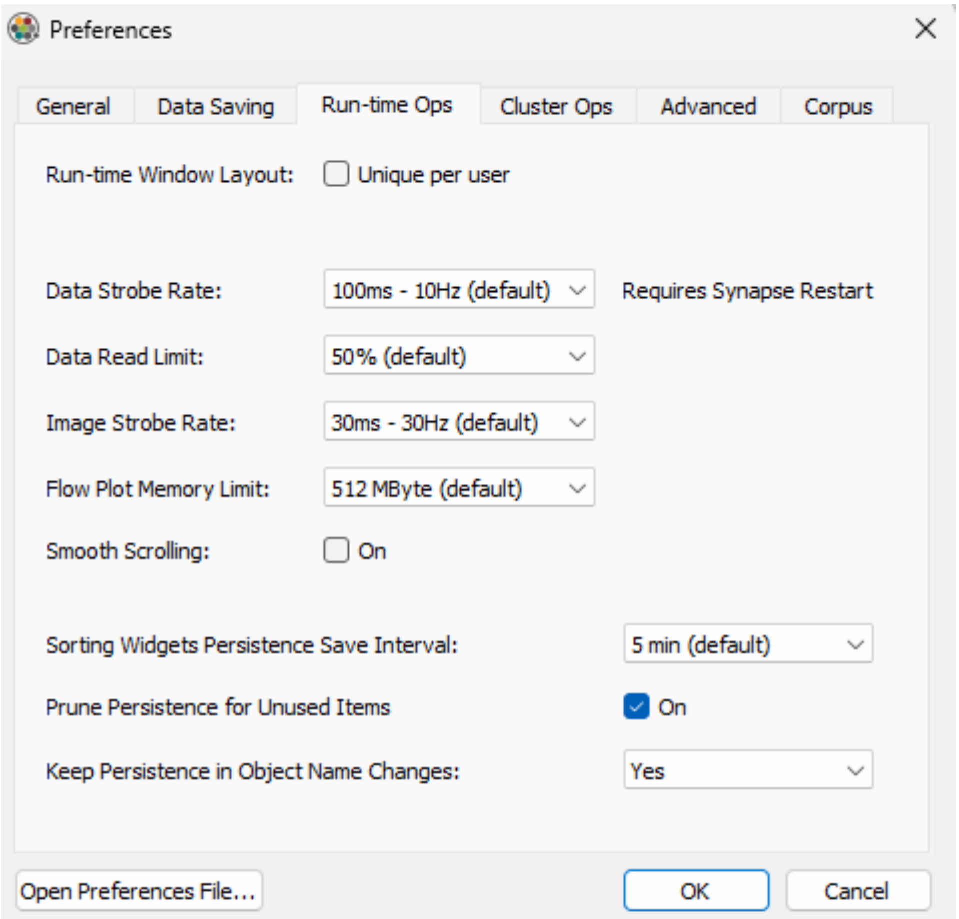
Data Saving



Preferences Dialog, Data Saving Tab

Option	Description
Hierarchy	Choose how tanks and blocks are associated and named. Experiments with Subjects - Experiments are the primary category under which data is stored. Subjects with Experiments - Subjects are the primary category under which data is stored.
Tank Naming	Choose how tanks are named. Auto - Tanks are named automatically based on preferences. New Tank Each Day - A new tank is created automatically each day. When NOT selected, the same tank is used until the user chooses to create or select a new tank. **Note:** When using New Tank Each Day, instead of including both data and time, Tank names include the date and Block names include the start time.
Path to Tanks	Enter or browse to choose a folder where tanks will be stored.
Block Naming	Choose how blocks are named. Auto - Blocks are named automatically based on preferences. Prompt - User is prompted to name each new block.

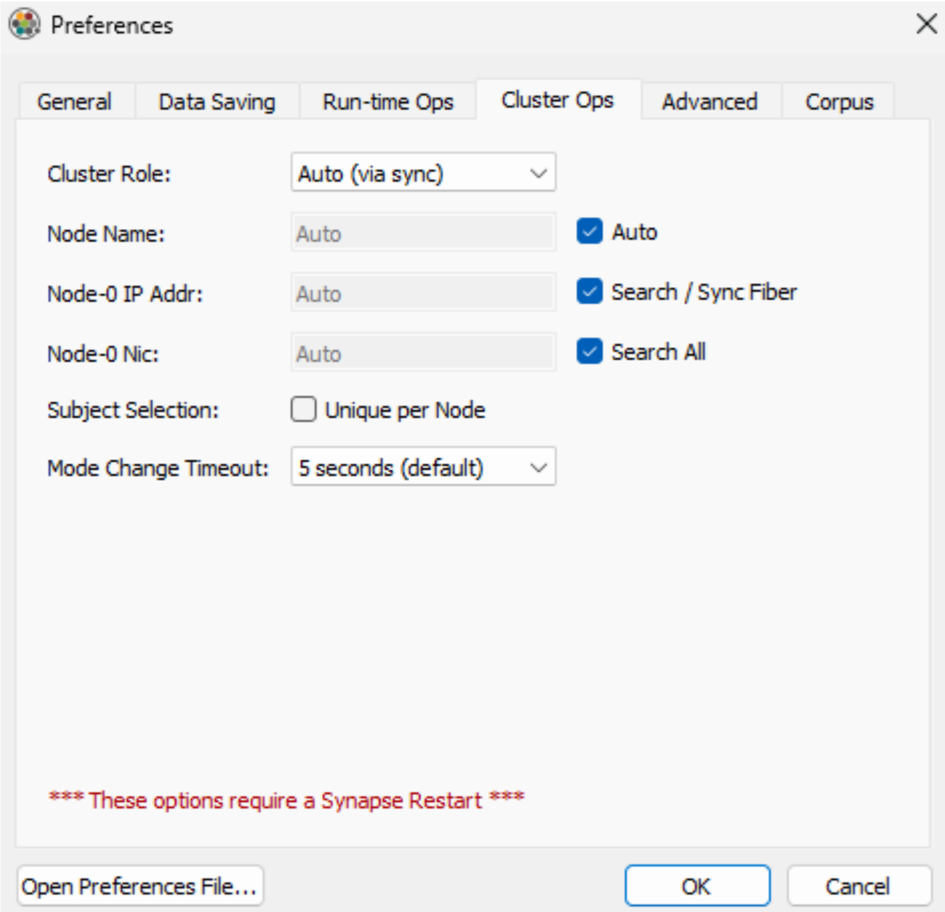
Run-time Ops



Preferences, Run-time Ops Tab

Option	Description
Runtime Window Layout	Save a unique window layout for each user
Data Strobe Rate	Choose how often data is saved to the tank and made available for plotting
Data Read Limit	Choose how much of the server resources are allocated to reading/storing data compared to distributing data to user interfaces. Increase this value if you are getting zBus limited readback warnings
Image Strobe Rate	Choose how often plots are visually updated
Scroll Memory Limit	Choose the maximum amount of memory available to the the main data plot time span and history
Smooth Scrolling	Enable Smooth Scrolling to improve the look of the Data Plot. Note that this adds a 1 second delay between when the data occurs and when it is shown in the Data Plot.
Sorting Widgets...	Choose how often the sort settings are logged into the database. Increasing this value reduces the amount of non-event related change data is recorded while setting up the sort codes.
Prune Persistence...	Removes persistence settings associated with gizmos that have been deleted from the Processing Tree. Also removes the persistence for channels in a sorting gizmo that are no longer used, for example when the channel count is reduced.
Keep Persistence...	When a gizmo name changes, you typically don't want to reset the persistence for that gizmo. Keep this set to Yes.

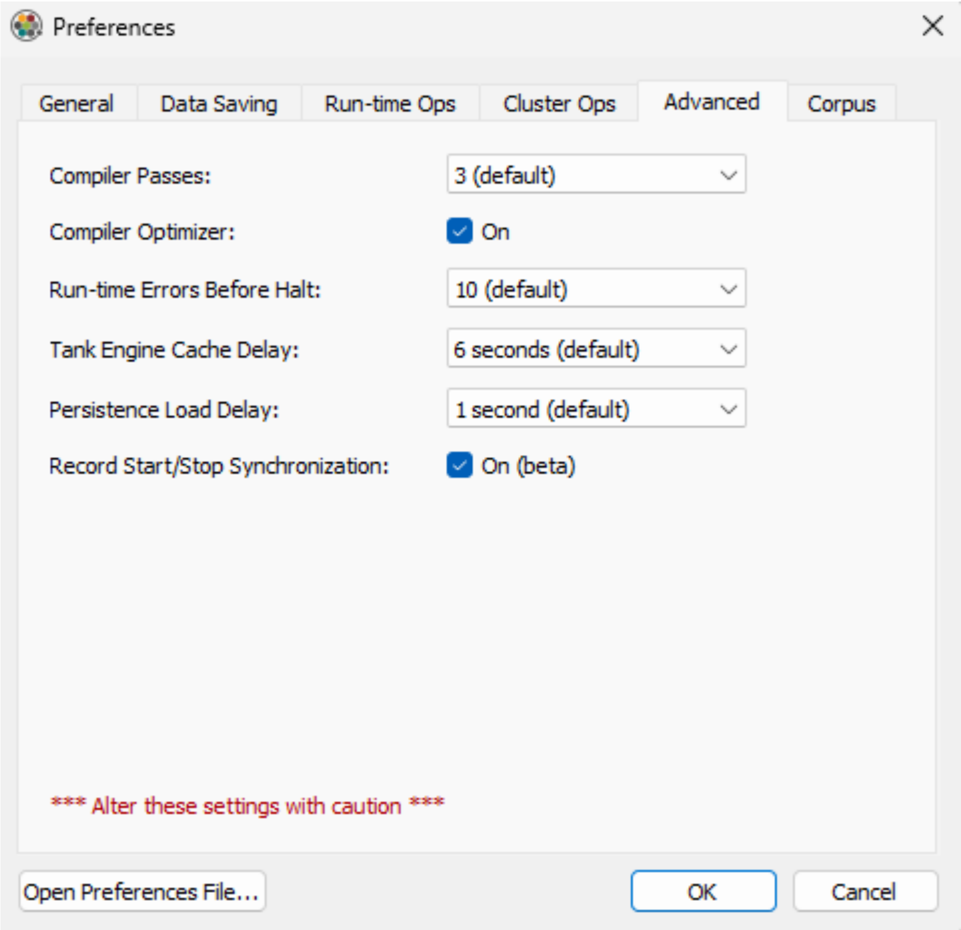
Cluster Ops



Preferences Dialog, Cluster Ops Tab

These settings are only used in Cluster Processing mode and are not available in Synapse Lite. See Cluster Preferences for more information.

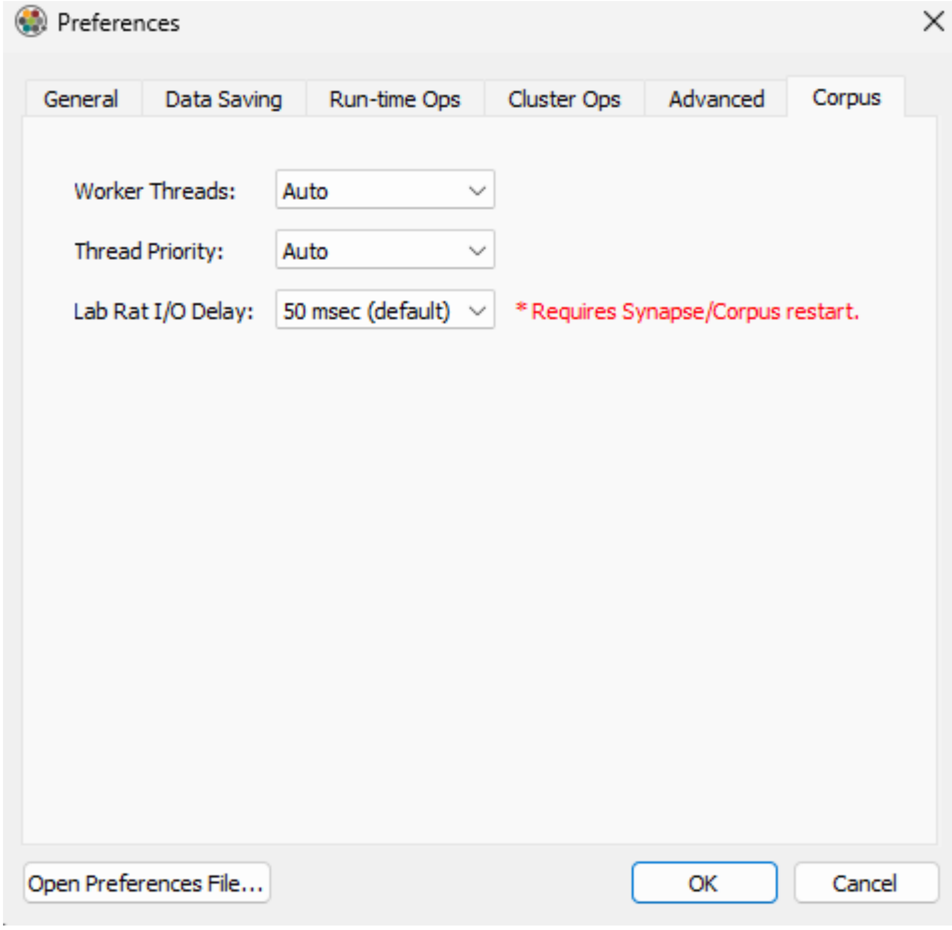
Advanced



Preferences Dialog, Advanced Tab

Option	Description
Compiler Passes	Set the number of attempts the compiler makes to compile the experiment. More passes may be required for larger and more complex experiments.
Compiler Optimizer	When checked, Synapse will remove any unused DSP components during compilation. Compilation will take longer, but this may be required for complex experiments to run on the current rig.
Runtime Errors Before Halt	Set the number of reported errors that can occur before Synapse switches to Idle mode. The count is reset each time Synapse is switched to Record mode.
Tank Engine Cache Delay	The size of the temporary memory buffers. Although a higher setting requires more system RAM, it makes data storage errors less likely.
Persistence Load Delay	Choose the amount of time allowed when switching from Standby to Preview. This ensures all the persistence settings are loaded onto the hardware before the experiment starts. Useful for more complex experiments.
Record Start/Stop Synchronization	Waits for all peripheral devices to be ready before beginning acquisition. Particularly useful for synchronizing the iV1/iV2 Video Capture Interface frame acquisition to the onset of the recording.

Corpus



Preferences Dialog, Corpus Tab

Option	Description
Worker Threads	Set the number of CPU threads allocated to Corpus. If you have any issues getting Corpus to run, try setting this to **1 Thread**.
Thread Priority	Set the priority for the Corpus application relative to other applications running on your PC, e.g. Synapse, MATLAB
Lab Rat I/O Delay	Determines the duration of the event loop running on Corpus and the round-trip delay between Corpus and the Lab Rat

Runtime

This section covers run modes, persistence in the Flow Plot, tank/block naming prompts, and log files.

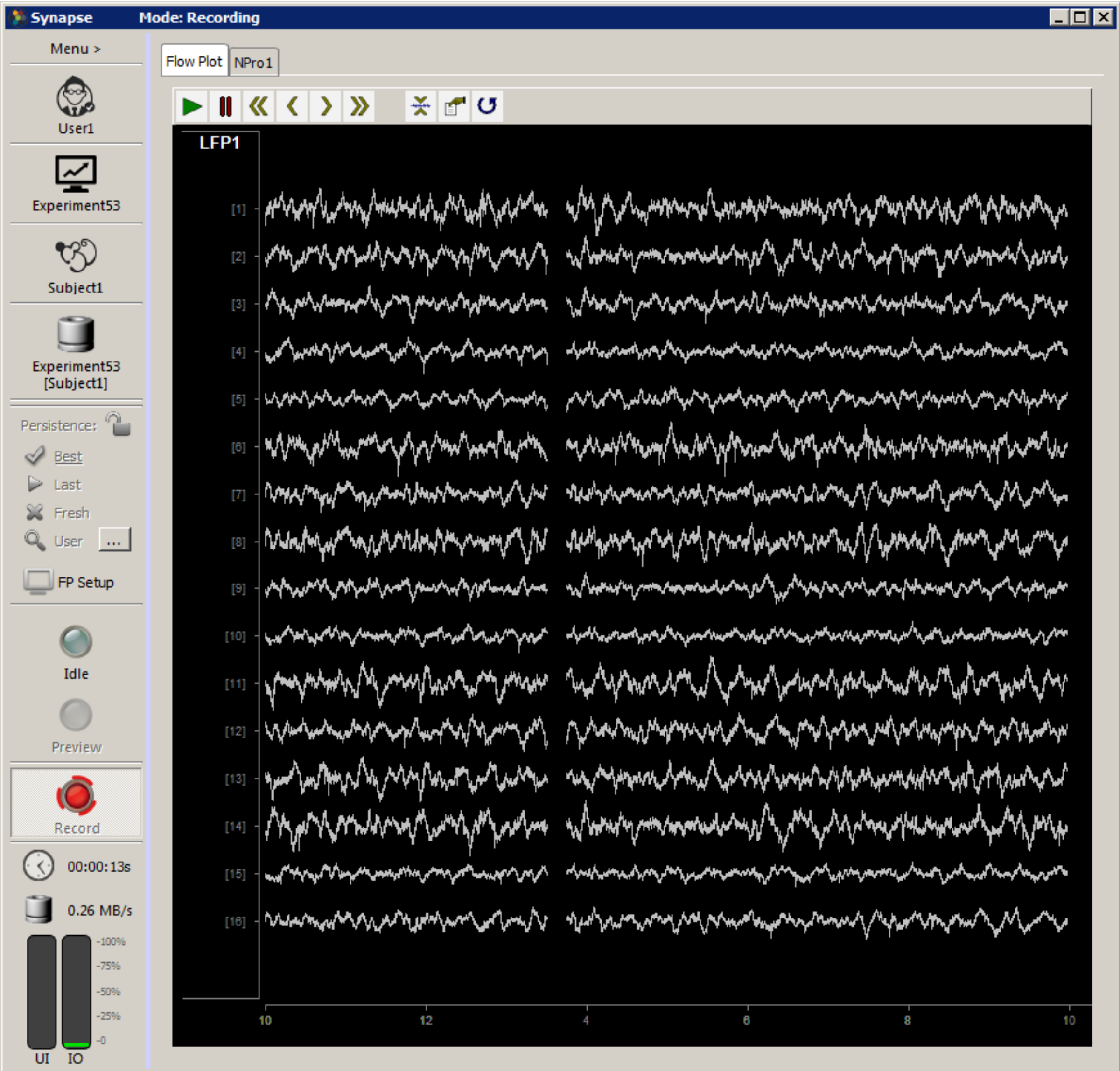
Controlling an Experiment

Runtime Modes

The control buttons allow the user to run or halt the experiment. They are enabled or disabled (grayed out) based on the available choices.

Mode	Description
Idle	Devices are not loaded and are not running.
Preview	Data is saved to a temporary block in the tank. Users can examine data in the Flow Plot. This allows users to modify parameter values before starting the experiment. Data is deleted when switching to Idle mode.
Record	Devices are loaded and running and data is saved to the tank. The Experiment and Subject must be configured before the Record button becomes active.
Standby	Disabled by default and enabled in the Synapse Preferences. In this mode devices are loaded and running but signals are not being acquired and saved to disk. Persistence values changed in Standby mode are saved to the database.

Flow Plot



Runtime Window

The Flow Plot window is automatically displayed in Preview and Record modes for fast, easy visualization of data.

The Flow Plot Tab

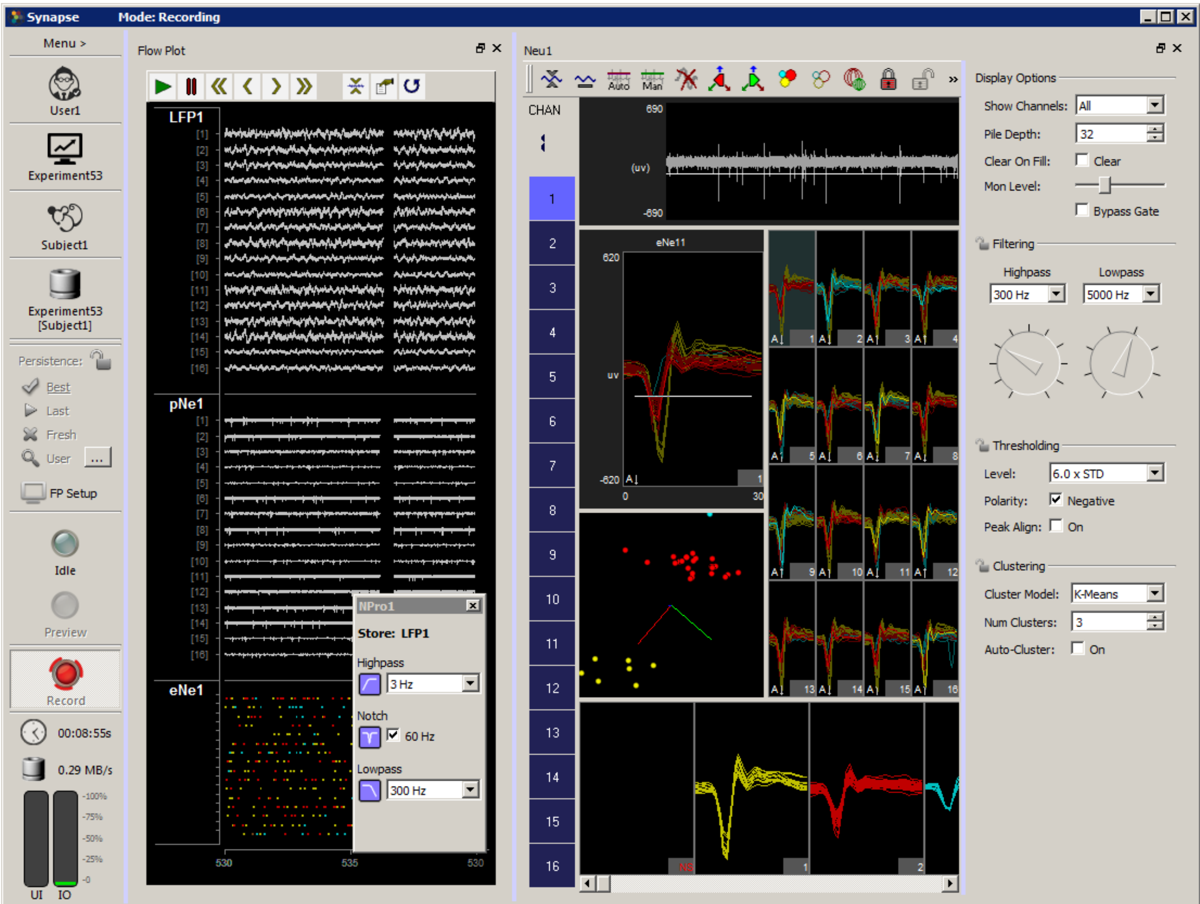
The primary plot includes a default plot configuration for each type of data being recorded. Users can adjust the plot settings to refine the display. The example plot above shows a subplot labeled LFP1 that displays 16 channels of streamed biological data using the Neural Stream Processor gizmo.

Other Tabs

Many gizmos add a tab to the tabbed window. The added tab contains runtime control features such as threshold and filtering controls.

Working with Tabs

Tabs can be floated, split, and merged back into the tab framework. Click and drag a tab to float and reposition the window. Right-click a tab then select an option on the shortcut menu to split the main window.



PCA Spike Sorting and Neural Stream Processor Interfaces

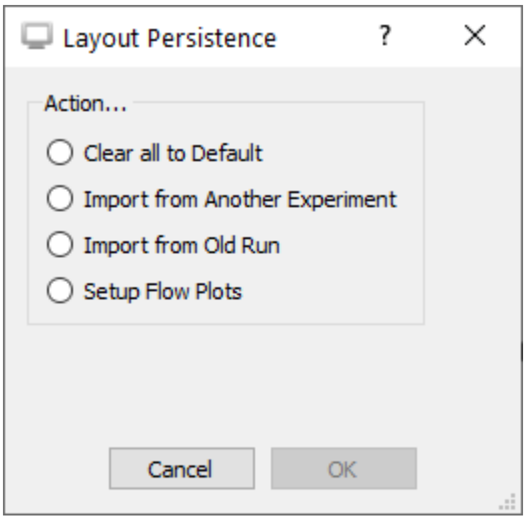
Persistence

Any changes to the run-time settings are saved with the Experiment and Subject in the Synapse database. This allows you to The Persistence options on the left toolbar determine how these run-time parameters are saved and reloaded.

Mode	Description
Lock	Locks the currently selected persistence. Otherwise the persistence selection always returns to **Best** when the experiment ends.
Best	Uses the last settings of any runtime controls for the current experiment and user/subject
Last	Use the last settings from the previous recording, regardless of subject or experiment. This is useful if you are using the same subject with a new experiment. For example, if you have PCA space sorting parameters defined for the subject and want to use them in a different experiment.
Fresh	Don't load any persistence, use the default experiment settings instead
User	Launch the History window. Right-click on a previous recording and choose "Use starting state" or "Use ending state", or right-click in the Changes list at the bottom to select a subset of changes from that recording to use for the next recording.
RT Layout	Choose the window layout configuration. See Layout Persistence below.

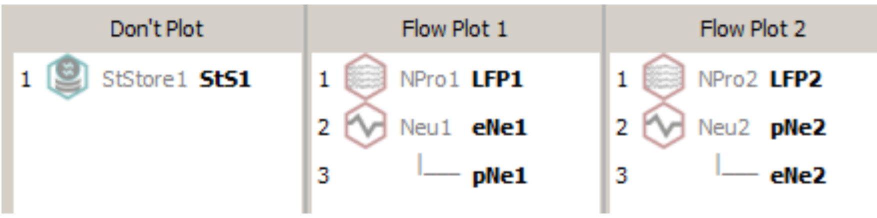
Layout Persistence

Information about the window layout and Flow Plot settings (scale factors, plot positioning, etc) is specific to the User and is separate from experiment persistence information. Options are accessible in the RT Layout button. You must have a named User for the layout to be saved.



Layout Persistence Dialog

Option	Description
Clear all to Default	Reset everything to the default layout and Flow Plot configuration scaling
Import from Another Experiment	Choose a previous User and Experiment combination from the Synapse database to load the layout from
Import from Old Run	Load the layout from a previous recording. This uses the block folder directly from disk and does not need the Synapse database for this.
Setup Flow Plots	Configure the ordering of the stores for the current experiment within the Flow Plot. This option also lets you split out different stores to entirely different Flow Plot tabs. This is useful for organizing the layout, particularly when doing multi-subject recordings. Right-click + drag a store name to another plot to make a copy.



Multi-Subject Example Layout

During run-time, the RT Layout button turns into an FP Setup button, which allows you to edit the location of the stores within the Flow Plot(s) and to hide them altogether.

Toolbar and Menu Reference

The Toolbar

A toolbar at the top of the Flow Plot allows the user to control plot animation.

The toolbar contains the following commands:

Icon	Description
	Play
	Pause
	Scroll back by plot window width (e.g. if span is set to 60 seconds, this button will scroll back in 60 second chunks)
	Scroll back (increments of span/10)
	Scroll forward (increments of span/10)
	Scroll forward by plot window width
	Auto Scale all plots
	Data Monitor Setup (launch dialog)
	Refresh

The Shortcut Menu

Additional commands for scaling and shifting plots are available from a right- click shortcut menu on the Store name in the plot.

Shortcut	Description
Auto Scale	Scale the display so that it best fits in the available subplot area.
Scale Up/Down	Incrementally scales the display up or down.
No Shift	Resets the y-axis center to 0.
Shift Up/Down	Shifts the display up or down in the subplot window.
Make larger/smaller	Makes the available subplot area larger or smaller. The other plots are resized accordingly.

Plot Display Options

Users can change the plot type, modify the number of channels viewed, and choose to color traces by channel or sort code in the Display Options dialog.

To modify the display options:

- Double-click the left side of the desired subplot, where the store name is displayed

Data Monitor Setup

Users can change settings related to the time span and tracking of the plot window in the Data Monitor Setup window.

Click the **Data Monitor** button on the plot toolbar to view data monitor settings for the plot.

Setting	Description
Time Span Control	
Span	Set the time span (sec) of the plot window.
History	Determine how much plot history will be available to view. Note how the memory requirements change as these settings are adjusted.
Tracking Mode	
Reference Epoch	If a reference epoch is selected, the left side of the Plot window will always coincide with the start of the reference epoch event.
Time Axis Overlap	Set the amount of the time axis that is repeated when the plot rolls over. For example, if the span is 10 seconds and Time Axis Overlap is set to 50%, the plot will show seconds 0-10, 5-15 etc.
Time Display Mode	Set the display units of the time axis.
More Settings	Press Shift + Ctrl and double-click the dialog box to display additional settings for the plot appearance, such as background color and labels.

Hardware Reference

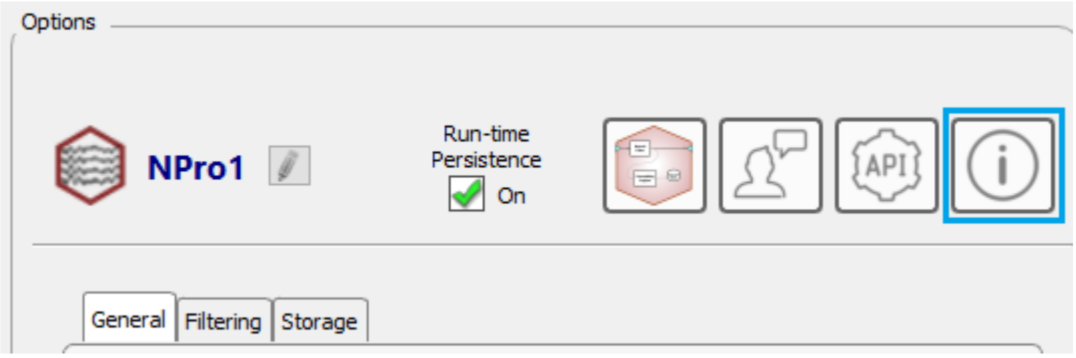
Hardware Reference

The hardware choices that you make in the Rig Editor appear as gray icons in the Processing Tree. Your experiment choices are mapped to the hardware in your rig and Synapse generates the required code instructions, automatically optimized for your hardware. Synapse also exposes the configuration options through an Options page for each piece of hardware. This gives you a streamlined way to make experiment-specific hardware choices, such as the number of recording channels, operational modes, and input sources.

Device settings for an experiment are displayed in the Options area of the designtime interface when the device is selected in the Processing Tree. Any change to hardware options must be committed or reverted by clicking the corresponding button in the Options area.

Getting Help From Within Synapse

You can access this manual directly within Synapse by clicking the information icon shown in the Options icon bar of any hardware object. This opens the web version of the Synapse Manual. If there is currently no internet access, it opens the local PDF and jumps you to the page for the hardware object you are currently looking at.



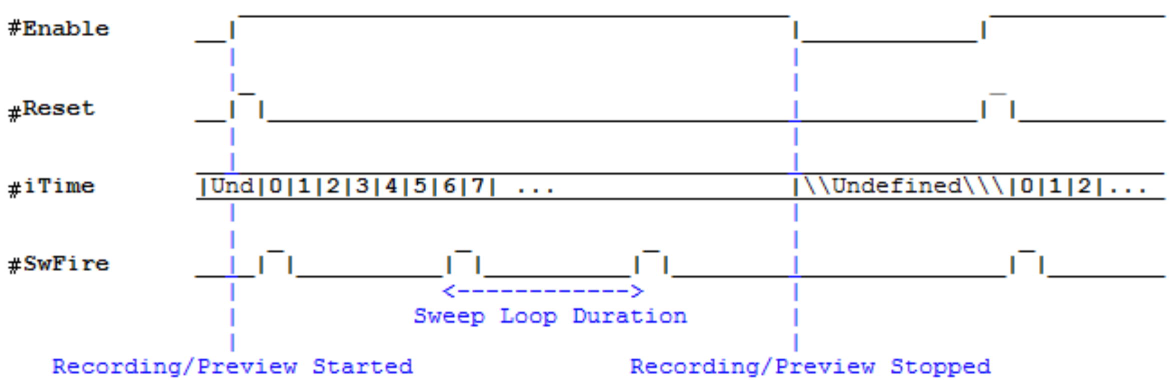
RZ and RX Processors

RZ Processor Options

Timing Signals

Before looking at the specific experiment HAL options, it is important to know that whenever an RZ Device is included in your Rig, the following timing signals will be available from the device. They will typically show up as drop down menu options when you are configuring a signal source for a gizmo.

Processor Signals	Type	Description
#Enable	Logic	Enables data storage through the duration of each block
#Reset	Logic	Resets counters and buffers at the start of each block.
#iTime	Integer	Samples elapsed since beginning of each block
#SwFire	Logic	A pulse that triggers once per second

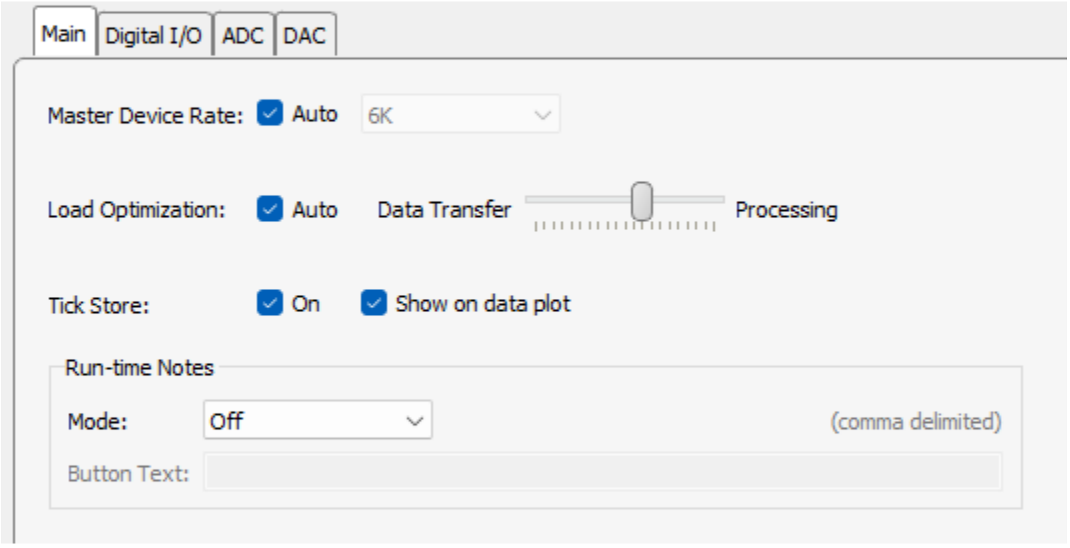


Time Control Waveform Diagram

RZ Options

Some options may not be available for some RZ devices.

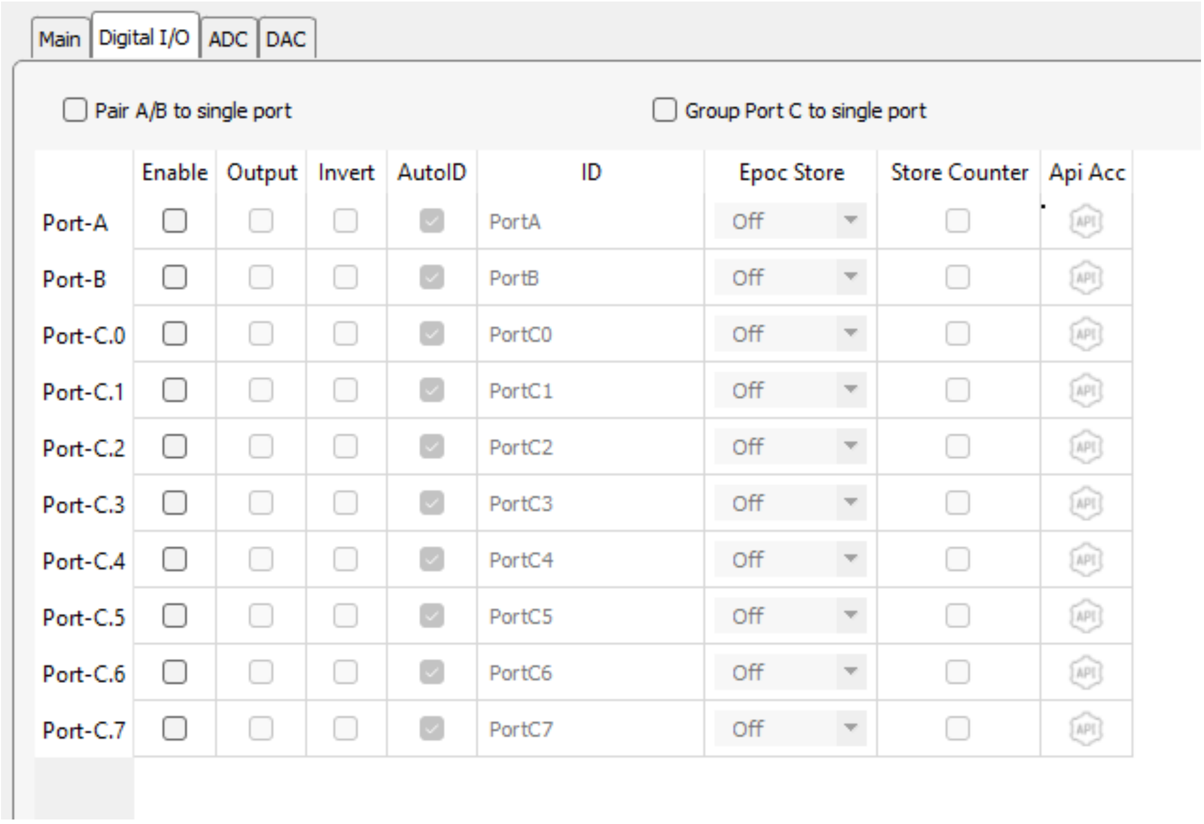
Main Tab



Main RZ Options Tab

Setting	Description
Master Device Rate	Select to allow Synapse to determine the master sample rate based on the gizmos assigned to this RZ, or clear it to enable the drop-down menu and set the sampling rate manually.
Load Optimization	Select to allow Synapse to balance the processing load between data processing and data transfer automatically. Clear to enable the slider and then drag the slider to set manually. For experiments that require heavy data transfer rates, move the slider to the left. For experiments with heavier processing loads, move the slider to the right. If you are all the way to the right, make sure the Compiler Optimizer is enabled in Menu → Preferences → Advanced.
Tick Store	Select to enable the Tick data store, which fires once per second. You can also choose whether to display this on the Data Plot during run time. The tank server that saves the data stops responding if there is a gap in the data that is greater than the Tank Engine Cache Delay (typically >6 seconds). The Tick store ensures there is always something saving into the tank. For higher demand applications you might not need/want the Tick store taking up processor usage. If there is continuous or regular data then there is no need for the Tick store.
Run-time Notes	Turn this feature on to record notes during run time that are associated with the current block. An additional run time tab lets you add custom notes. These will be included in the database and will be available as a text file inside the block folder. If 'Notes File + Epocs' is selected, a timestamp of when the note occurred along with a value code will be included in the data tank in the 'Note' epoc store. **Note:** This only works In Record mode and is not available in Preview mode.
Button Text	If you want to quickly mark events as they are happening during the recording, add a comma separated list of the notes you want. This text will appear on individual buttons that will add the note when clicked.

Digital I/O Tab



Digital I/O Options Tab

The RZ has 24 total bits of digital I/O, configured in two bytes (Port-A and Port-B) and eight bits (Port-C). Enable the desired I/O and set the direction with the **Output** check box. When the **Output** check box is selected for a given row, a data source must be selected in the ID column. If the **Output** check box is cleared, that row turns into a data source that can be connected to other gizmos. The name of the data source is set in the ID column.

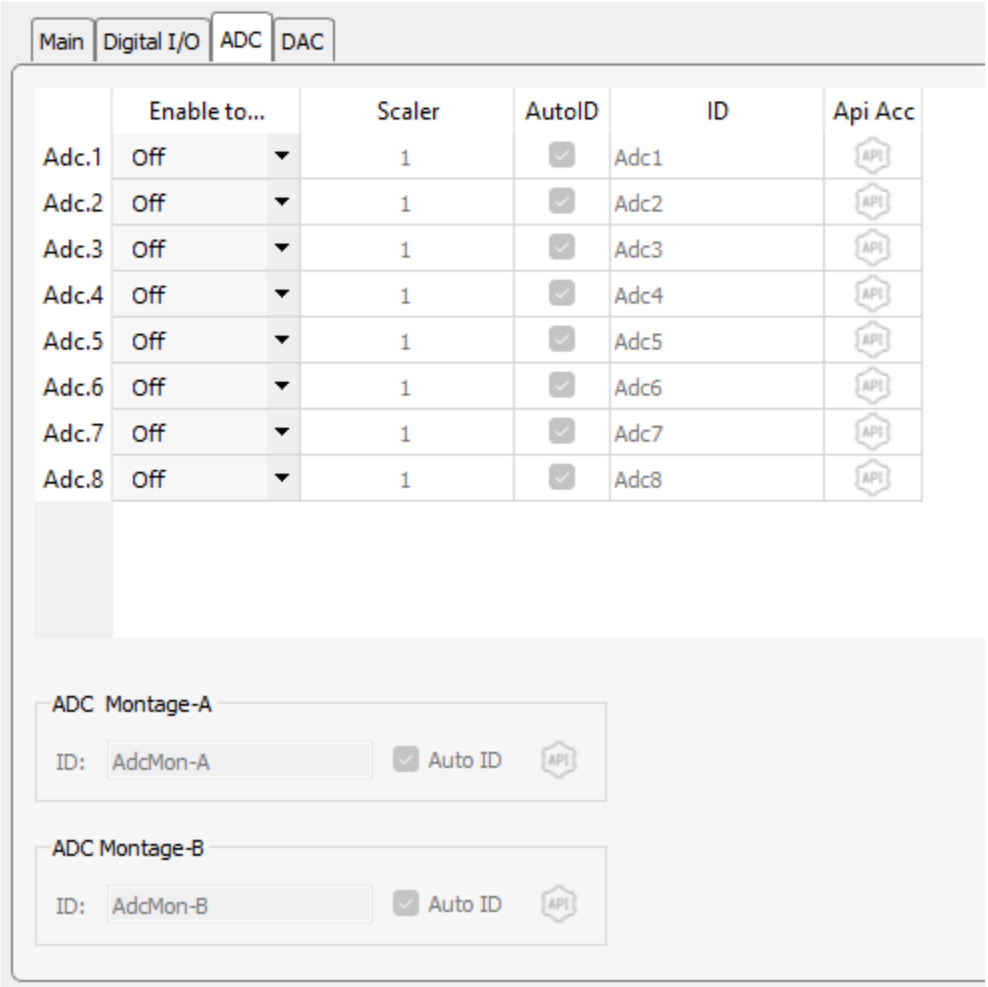
A counter may also be stored for bits in port C. The counter may be useful for synchronizing to an external camera system.

Select the **Pair A/B to single port** check box to combine A and B into a single 16-bit integer source or input link.

Select the **Group Port C to single port** check box to combine the eight bits into a single byte.

Set **Epoc Store** to 'On Change' to automatically add an Epoc Store for the corresponding byte or 'Full', 'Onset', or 'Offset' for corresponding bit. When this method is used, you don't have to add an Epoch Event Storage gizmo to the processing tree.

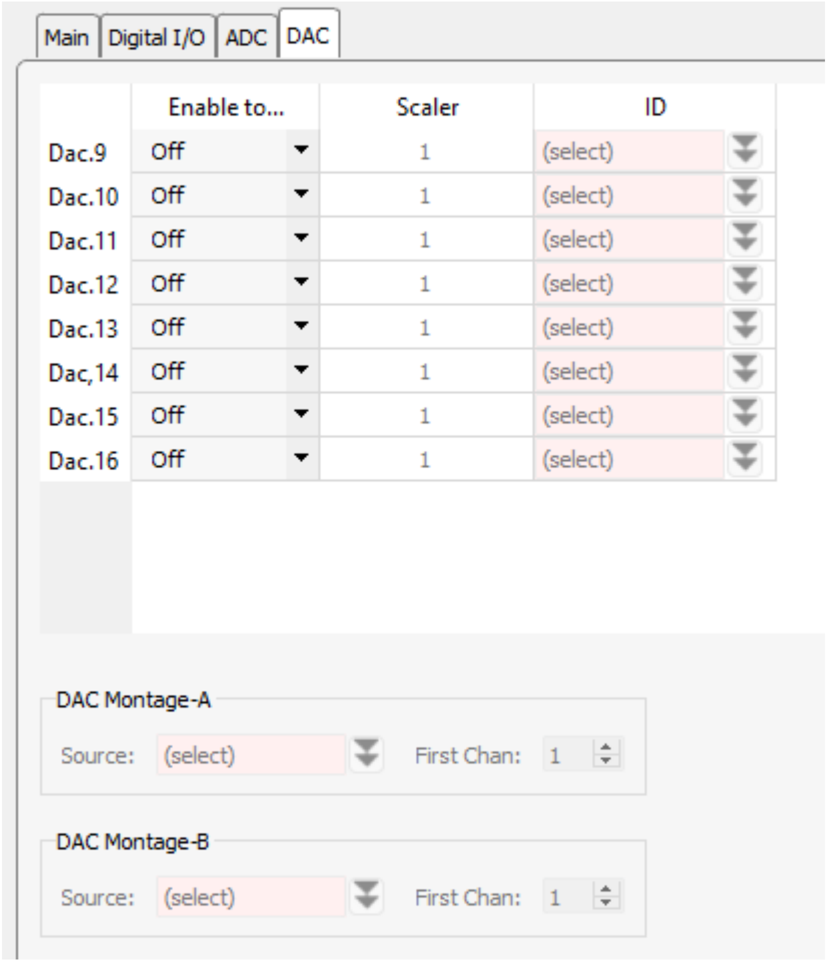
ADC Tab



ADC Options Tab (shown for RZ2)

Analog input channels appear on the ADC tab. Each channel can be used individually as a single channel floating point data source for other gizmos, or can be grouped into one of two montages, which are multi-channel floating point data sources. You can also apply a scale factor to each channel to convert to the correct units.

DAC Tab

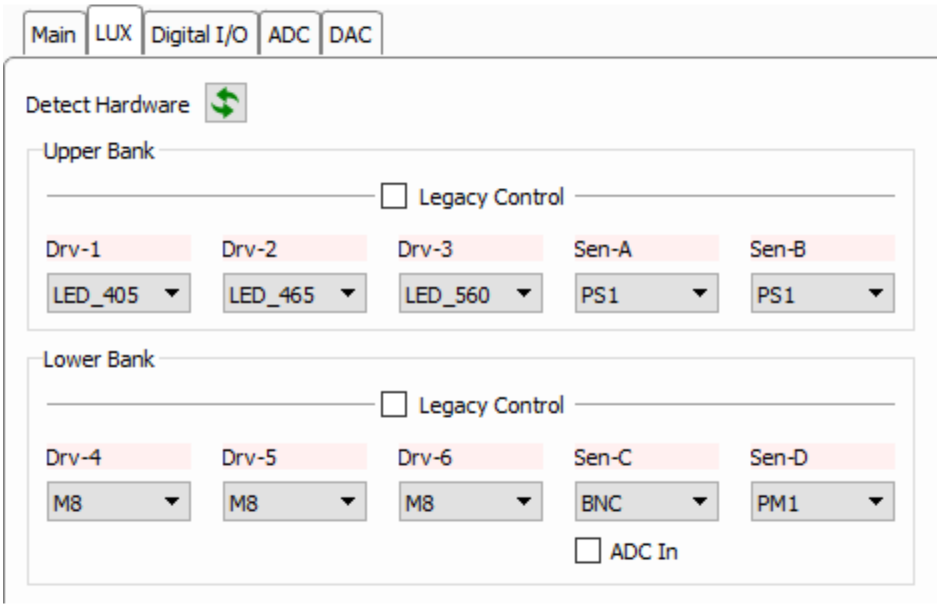


DAC Options Tab (shown for RZ2)

Analog output channels appear on the DAC tab. For RZ6, the built in attenuators also appear on the DAC tab.

Each channel can be used individually as a single channel floating point data sink, or can be grouped into one of two montages, which are multi-channel floating point data sinks. You can also apply a scale factor to each channel to convert to the correct units before it is sent out of the RZ. A data source for the enabled output channels/montages must be selected.

LUX Tab (RZ10x only)



LUX Options Tab

The RZ10x processors have LUX light drivers, photosensors (PS2), and light power meters (PM1) built into the front panel. They are configurable with slots for up to 3 drivers and 2 sensors/power meters per bank. Each RZ10x has two banks.

With the RZ10x powered on, click 'Detect Hardware' to automatically read the LUX component configuration into Synapse. This will inform gizmos that communicate with the LUX components, like the Fiber Photometry gizmo, about the components that are available.

Built-in LUX LEDs are named by their nanometer wavelength. To use an external LED, use an M8 connector in the light driver slot instead. To use an external LED driver or generic DAC output, a BNC connector is placed in the light driver slot. The BNC port can be accessible as a standard DAC signal output, available in the DAC tab, by checking the 'DAC Out' checkbox below the driver.

If a BNC port is in a sensor slot, it can be accessible as a standard signal input in the ADC tab by checking the 'ADC In' checkbox below the sensor.

Legacy Control

The Legacy Control option makes the entire LUX bank available on the ADC and DAC tabs as regular gizmos inputs/outputs. Use this if you are driving the LEDs with your own custom stimulus signal and not controlling it with one of the gizmos that are specifically designed to target the LUX banks, e.g. Fiber Photometry.

The LUX LED drivers have three output ranges that are exposed in this mode: 200 mA, 500 mA, and 1000 mA. Choose the option that best suits your target output to get the best resolution.

The output range is relative to a 10 V driving signal, so make sure your control signals are scaled properly to get your desired output. For example, a Pulse Train Generator gizmo outputting a 5 V signal to a LUX driver that is set to 500 mA range will output a 250 mA signal to the LED.

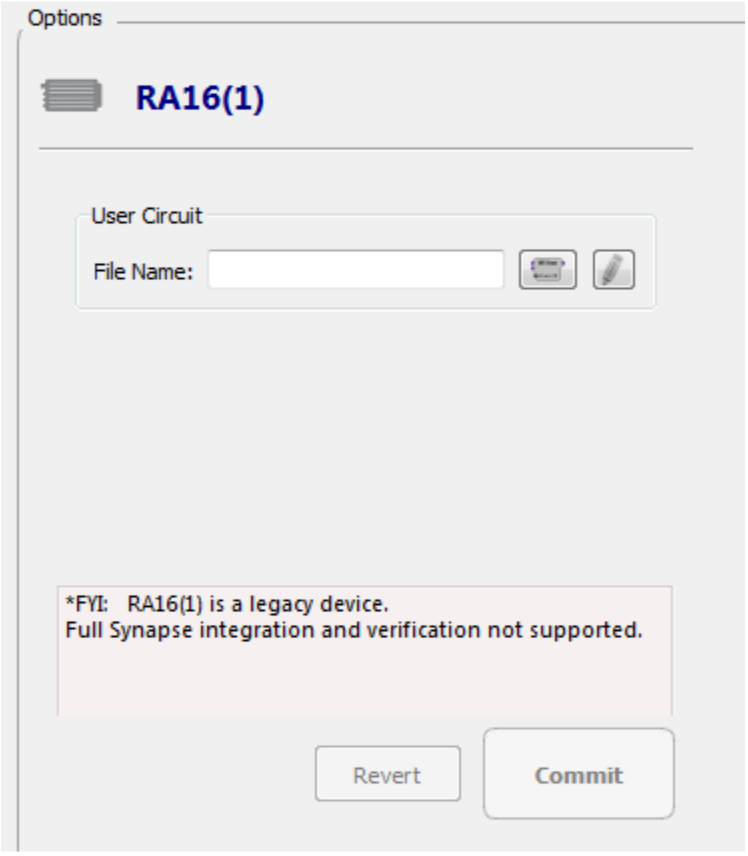
RX Processors

See RZ Processor Options above. There are additional options in the Digital I/O tab to determine which port the front panel lights on the RX display.

Legacy Mode

Legacy mode can be used to directly load experiment circuit files (*.rcx) to System 3 processors. This feature allows customers who are transitioning from OpenEx or user-developed TDT applications to port existing experiments directly into Synapse.

The RP2.1, RA16, and RX7 can only be used in legacy mode. Other processors can be switched into legacy mode in the Rig Editor.



Legacy Mode Device Options

The illustration above shows the legacy device options for the RA16 Medusa Base Station. The Options are the same for legacy mode, regardless of the device.

Note

Parameter tags in the legacy *.rcx file can be accessed using SynapseAPI.

You can also use gizmoControl macros within Legacy HALs to add widgets to the user interface.

Any storage macros in the legacy circuit will also plot on the Flow Plot.

Options

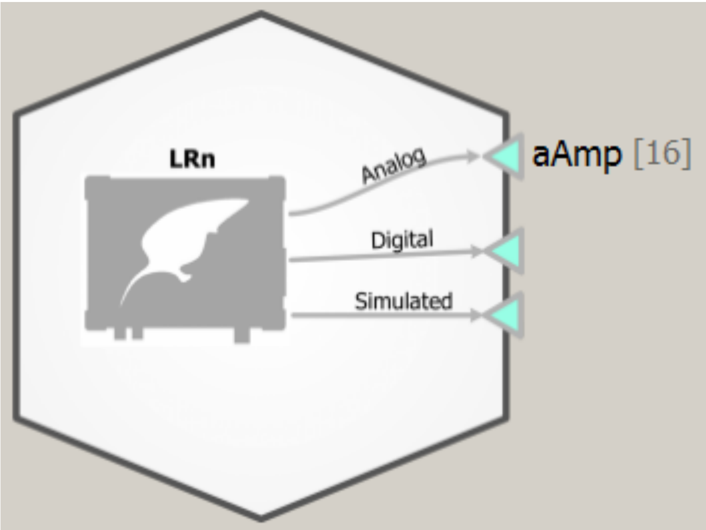
File Name Enter the path and file name of the circuit to load.

Select Circuit File button Launches an Open window. Select the desired file.

Edit Circuit in RPvdsEx button Launches RPvdsEx. Edit or create a circuit file.

LR10 Lab Rat Interface Module

The Lab Rat HAL is the nexus for all interactions with the Lab Rat interface module. The Lab Rat is only available in Synapse Lite software. The LR10 HAL is an independent object in the Processing Tree. However, the acquisition processing and I/O functionality are internally dependent on Corpus software and emulated RZ2 hardware.



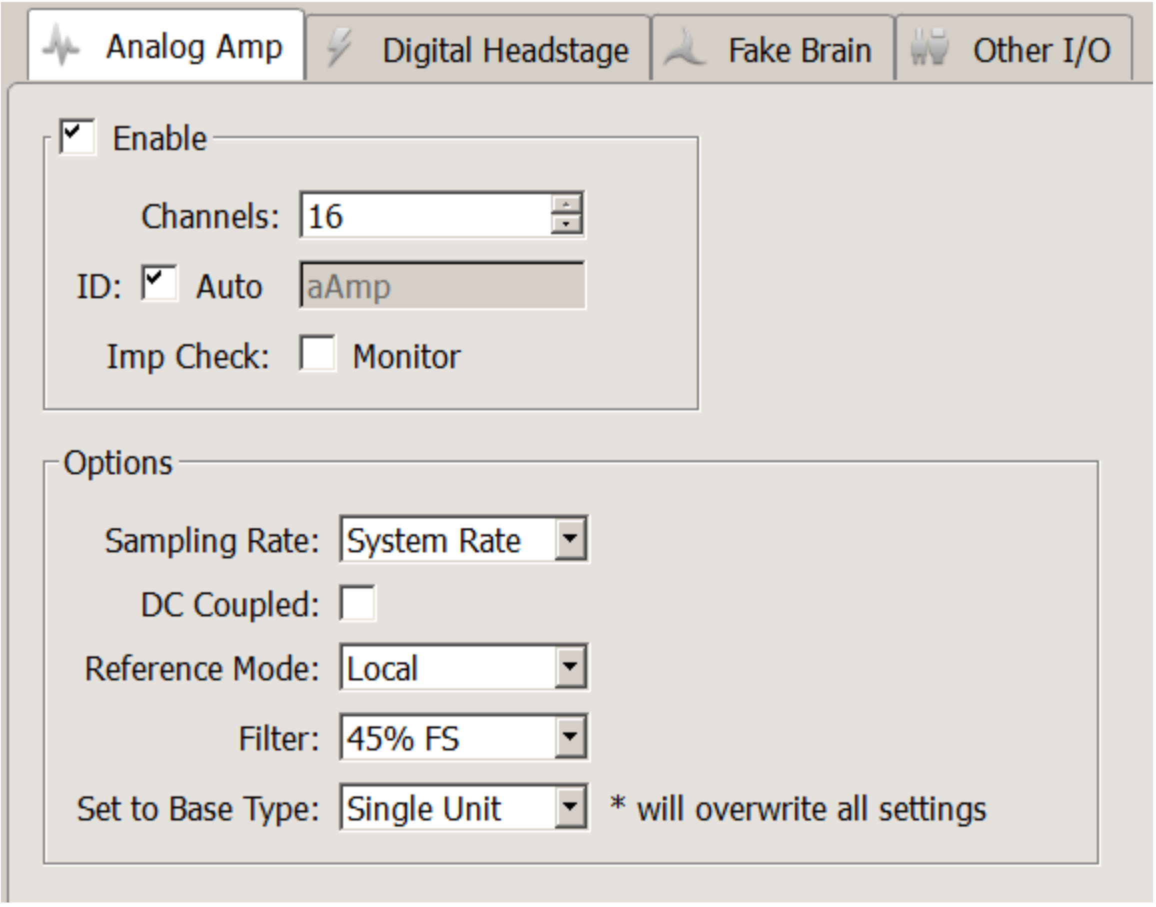
LR10 Block Diagram, showing the analog amplifier enabled

LR10 Options

Controls for analog and digital amplifiers, synthetic neural data generation for experiment testing, and multi I/O communications are modified within the Lab Rat object. Here, subcomponents can be enabled or disabled, and their settings can be adjusted prior to experimental run time. The four tabs of the LR10 are: Analog Amp, Digital Headstage, Fake Brain, and Other I/O.

Analog Amp

The Analog Amp page controls the 16-channel analog neural input on the Lab Rat interface module. From this page, the user can: enable/disable the amplifier, control how many channels of input to read, and adjust the sampling rate and filter settings of the amplifier. When the amplifier is enabled, the amplifier ID will appear as a Gizmo output at the top of the HAL and can be connected to other gizmos for further signal processing, with your selected channel count and amp ID.

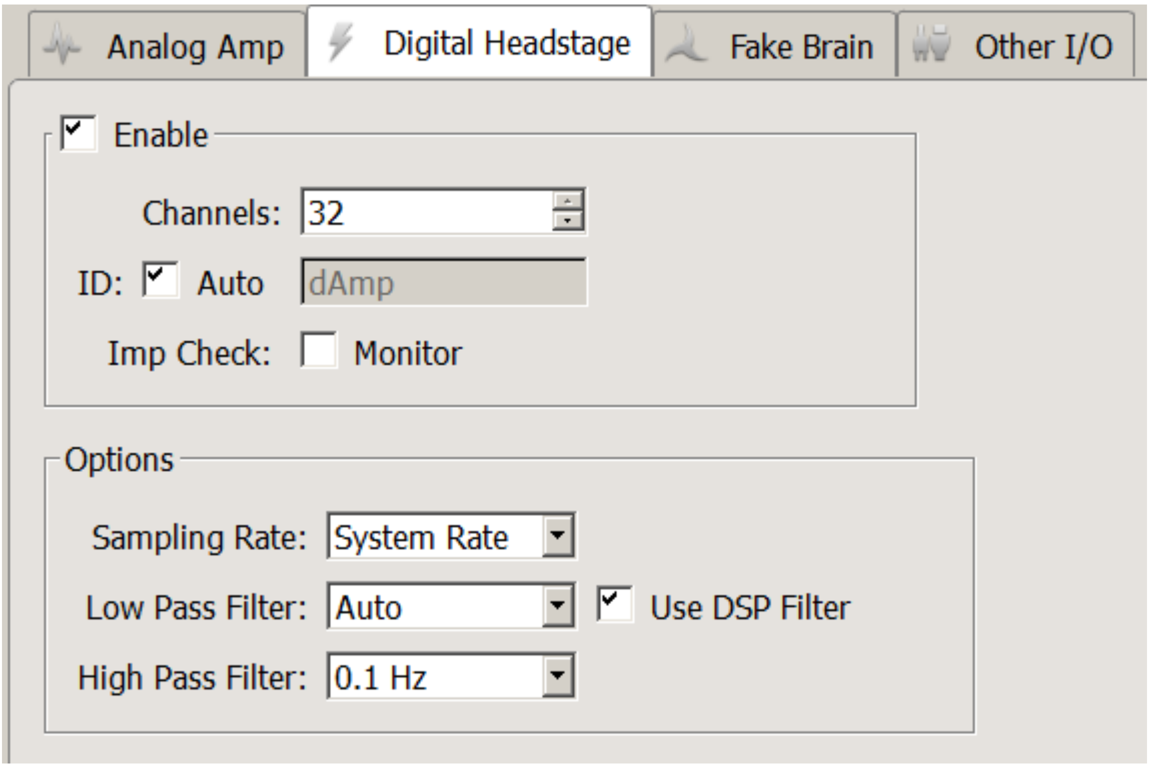


LR10 Analog Amp Options

Option	Description
Imp Check	Show online impedance testing results for the connected electrodes. See Runtime Interface.
Sampling Rate	Set the Sampling Rate to match the desired frequency band of your incoming signals (or leave at 'System Rate' if you are unsure). By default, the sampling rate matches that of the emulated RZ2.
DC Coupled	Remove the 0.4 Hz high pass filter on the input signals and record DC potentials.
Reference Mode	Select the mode from the drop-down menu. Local - all channels use a single reference (pin 5 on the DB26). Differential - each even channel acts as a reference for the odd channel before it. Note: the output channels will be mapped for you to remove duplicate channels.
Filter	Set the anti-aliasing low pass filter cutoff as a percentage of the sampling rate.
Set to Base Type	If you are unsure, use Set to Base Type to configure the amp with default settings based on common signal types.

Digital Headstage

Enable the Digital Headstage when using an Intan-based digital headstage.



LR10 Digital Headstage Options

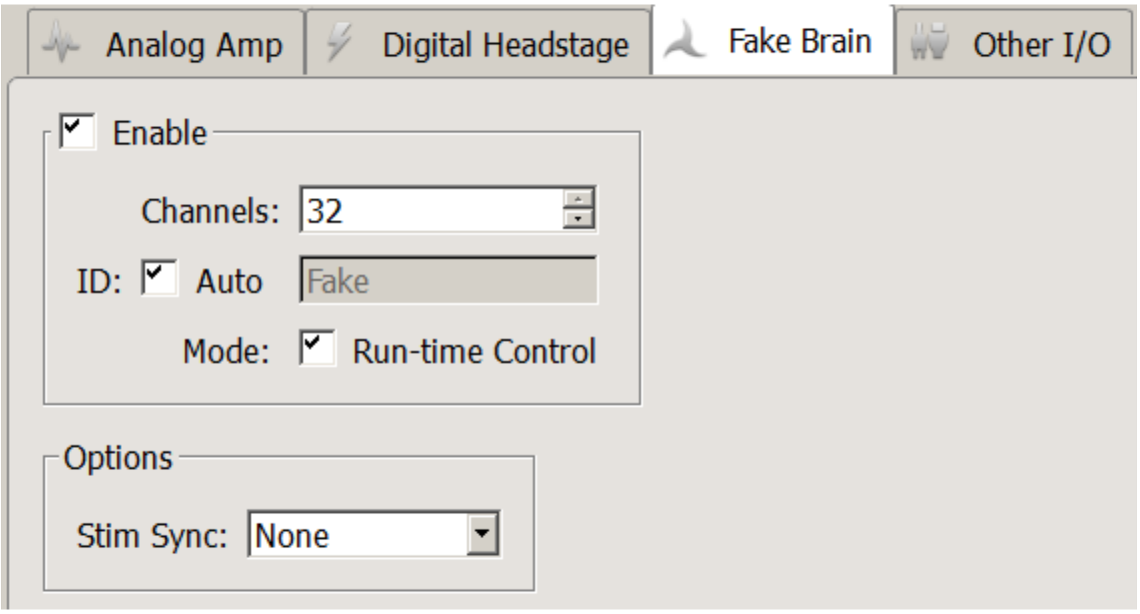
Option	Description
Imp Check	Show online impedance testing results for the connected electrodes. See Runtime Interface.
Sampling Rate	Set the Sampling Rate to match the desired frequency band of your incoming signals (or leave at 'System Rate' if you are unsure). By default, the sampling rate matches that of the emulated RZ2.
Low Pass Filter	Choose a value for the low pass filter implemented on the digital headstage chip. If Low Pass Filter is set to "Auto," the cutoff frequency will be configured according to the conversion table below.
Use DSP Filter	Adds additional low pass filtering (performed by Corpus), matched to the selected Low Pass Filter frequency. This removes high-frequency digital noise that is added by the digital headstage chip.
High Pass Filter	Implemented on chip.

'Auto' low pass filter settings for different sub amp sampling rates

Digital Amp Sampling Rate	LP Auto Filter Cutoff
750 Hz	300 Hz
1.5 kHz	750 Hz
3 kHz	1.5 kHz
6 kHz	3 kHz
12 kHz	5 kHz
25 kHz	10 kHz

Fake Brain

The Fake Brain is a synthetic data generator. This is a useful tool for designing experiments with well-behaved signals. The Fake Brain acts like an amplifier, and streams generated signals from the Lab Rat into Corpus. This means that you can connect Gizmo inputs to the Fake Brain output. Please take care to double-check that you are not handling and saving synthetic data during real experiments.



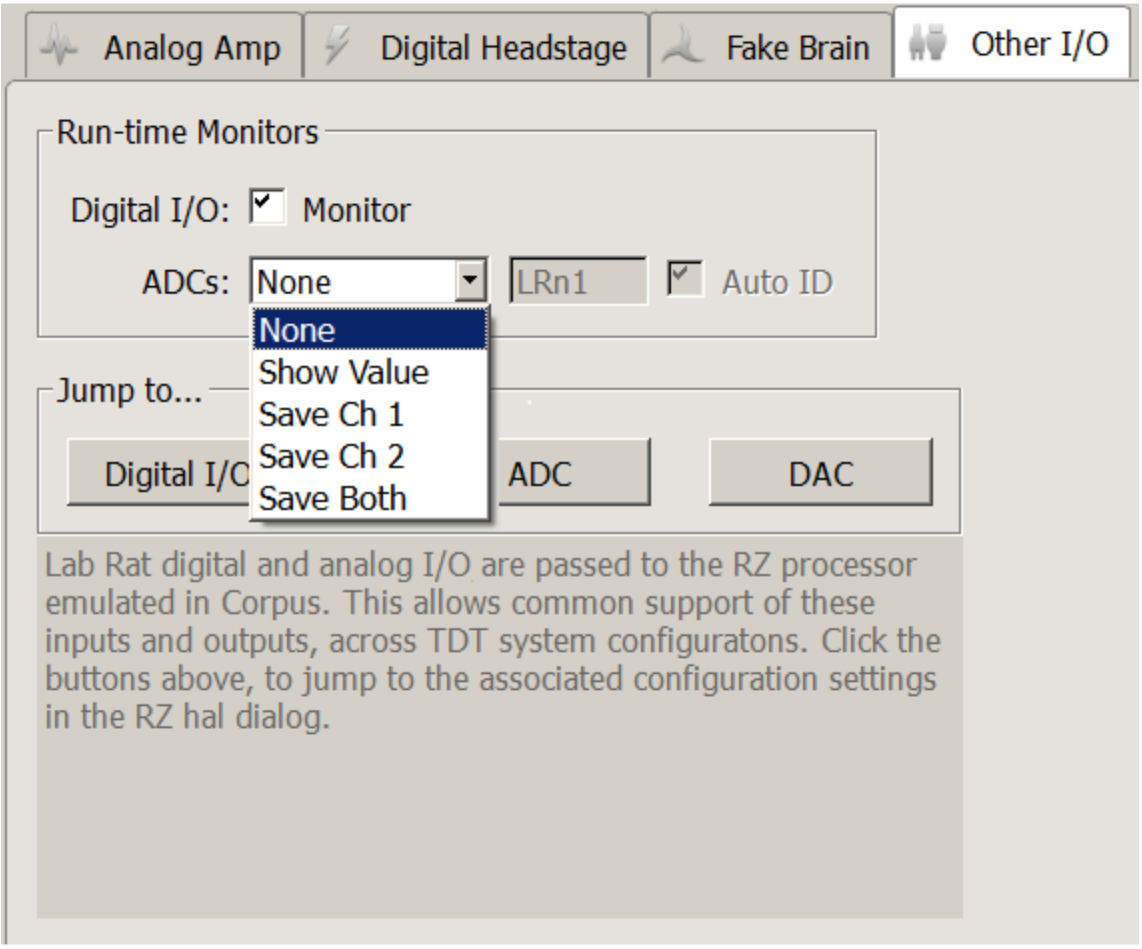
LR10 Fake Brain Options

Option	Description
Mode	Enable the Run-time Control to change the type of generated waveforms dynamically. These include: **Normal**: default mode, includes LFP and spikes. **Hash**: like Normal, but spikes reduced by a factor of 2. **LFP Only**: like Normal but no spikes. **Tetrode**: like Normal, but spikes on each group of four channels fire synchronously. **Sync 100 Hz**: spikes fire at 100 Hz on all channels. **Tone 30 Hz**: A 30 Hz ~700 uV sine wave on all channels. **Tone 1 kHz**: A 1 kHz ~70 uV sine wave on all channels. **Tone Ref**: A 100 Hz ~700 uV sine wave on all channels.
Stim Sync	Simulate inhibitory or excitatory input to the fake spike modes (Normal, Hash, Tetrode) with the Stim Sync option. You can enable this with an external device input, like the Bit0 BNC connector on the front of the LR10, to modulate firing rate with external hardware. When the Stim Sync input is activated, certain spike shapes are inhibited (fire less) and others are excited (fire more). To control the stim sync using only the LR10, use the Gizmo Input option.

Note

See the FB128 section of the System 3 Manual for more information on the Fake Brain modes.

Other I/O Tab



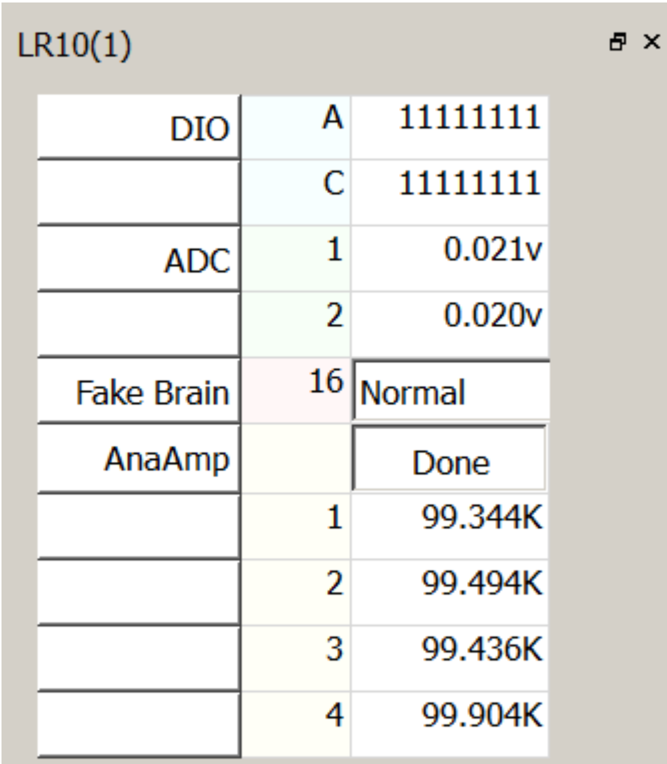
LR10 Other I/O Tab

Option	Description
Digital I/O Monitor	Shows you the logic state of all digital I/O on the Lab Rat at run time.
ADCs	You can choose to save the analog inputs coming into the "A/D" connector to disk or just view them during your experiment.

Lab Rat digital and analog I/O are passed to the emulated RZ2 processor in Corpus. This allows common support of these inputs and outputs across TDT system configurations. Software configuration of the multi I/O parameters is controlled in the RZ HAL. The Jump to... section takes users to the appropriate RZ HAL page for I/O configuration. For more information, please overview See RZ Options..

Option	Description
Digital I/O	16 bits of digital input (8 bit-addressable, 8 word-addressable) are available.
ADC	Two channels of ADC input are available as single-channel inputs or combined into a multi-channel signal for further processing.
DAC	Two channels of DAC output are available as single-channel outputs or can be combined and controlled by a multi-channel signal. Each DAC channel can be tied to the output of a Gizmo, such as the Electrical Stimulation or File Stimulation gizmos.

Runtime Interface



LR10 Runtime Interface

When **Digital I/O Monitor** is enabled on the **Other I/O Tab**, the state of all 16 bits is shown on the runtime interface in binary format.

If **Show Value** is selected for the ADCs option on the **Other I/O Tab**, the current voltage on the ADC inputs will be shown at runtime.

If **Run-Time Control Mode** is enabled in the **Fake Brain Tab**, you can change the fake signal type dynamically.

If the **Imp Check** option is enabled for the analog and/or digital headstage, use the interface button to switch the LR10 into impedance checking mode. The current impedance values of the connected electrode will update on the user interface. During the impedance check, the signals returned from the amplifier will be the value of the impedance, in MOhm. If the impedance of an electrode is above 500 kOhm, the monitor will return a value of "High Imp."

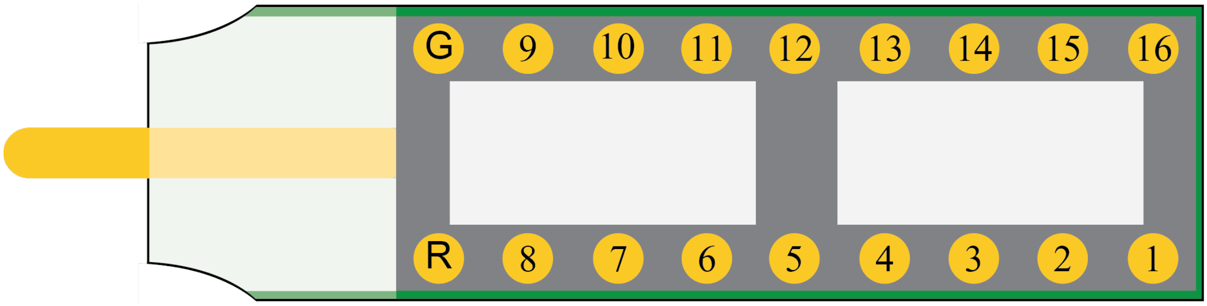
The impedance check uses a 90 nA RMS sine wave signal. The frequency depends on the sampling rate of the sub-amplifier.

Analog Amp Sampling Rate	Impedance Check Frequency
750 Hz	35 Hz
1.5 kHz	70 Hz
3 kHz	140 Hz
6 kHz	280 Hz
12 kHz	560 Hz
25 kHz	1120 Hz
Digital Amp Sampling Rate	**Impedance Check Frequency**
750 Hz	24 Hz
1.5 kHz	48 Hz
3 kHz	95 Hz
6 kHz	191 Hz
12 kHz	381 Hz
25 kHz	763 Hz

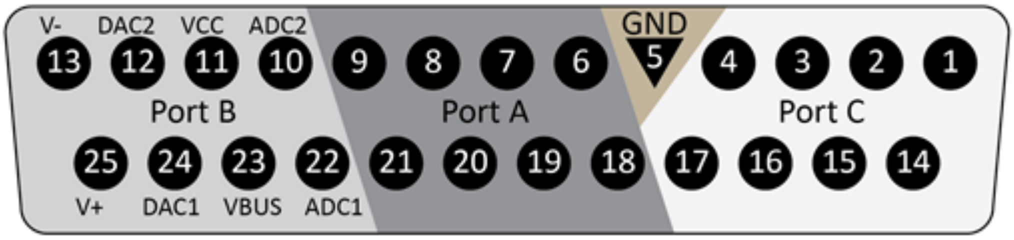
Pinouts

The pinouts are looking into the connector.

AC16LR Headstage

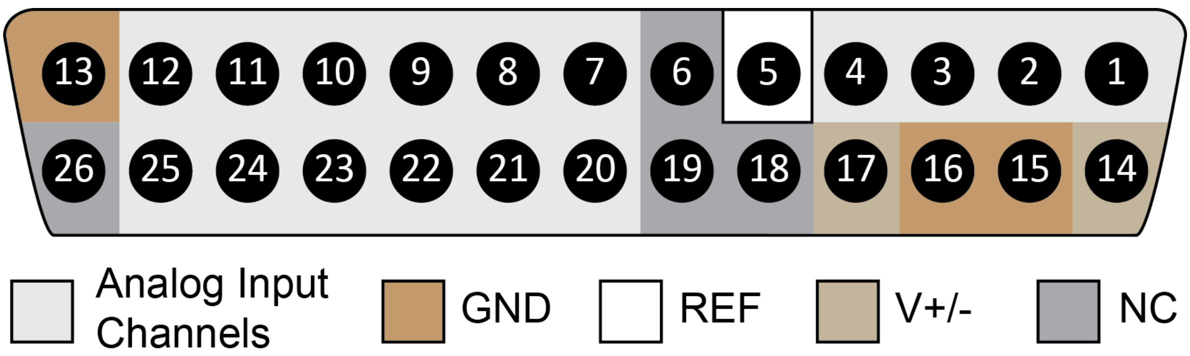


DB25 Connector



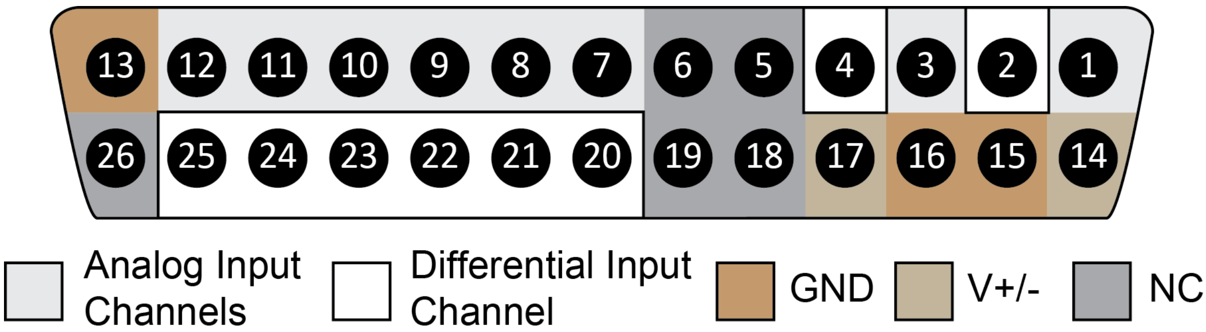
Pin	Name	Description	Pin	Name	Description
1	C0	Port C Bit Addressable	14	C1	Port C Bit Addressable
2	C2	Digital I/O	15	C3	Digital I/O
3	C4	Bits 0, 2, 4, 6	16	C5	Bits 1, 3, 5, 7
4	C6	+3.3 V, 10 mA max per bit	17	C6	+3.3 V, 10 mA max per bit
5	GND	Ground	18	A0	Port A Word Addressable
6	A1	Port A Word Addressable	19	A2	Digital I/O
7	A3	Digital I/O	20	A4	Bits 0, 2, 4, 6
8	A5	Bits 1, 3, 5, 7	21	A6	+3.3 V, 10 mA max per bit
9	A7	+3.3 V, 10 mA max per bit	22	ADC1	Analog Input Channel 1 ±3 V, 20 mA max
10	ADC2	Analog Input Channel 2 ±3 V, 20 mA max	23	VBUS	+5 V, 150 mA max
11	VCC	+3.3 V, 250 mA max	24	DAC1	Analog Output Channel 1 ±3 V, 20 mA max
12	DAC2	Analog Output Channel 2 ±3 V, 20 mA max	25	V+	+3.3 V, 150 mA max
13	V-	-3.3 V, 150 mA max			

DB26 Neural Input - Shared Reference Mode



Pin	Name	Description	Pin	Name	Description
1	A1	Analog Input Channels	14	V+	Positive Voltage (+2.5 V)
2	A2		15	GND	Ground
3	A3		16	GND	Ground
4	A4		17	V-	Negative Voltage (-2.5 V)
5	Ref	Reference	18	HSD	Headstage Detect
6	HSD	Headstage Detect	19	HSD	
7	A5	Analog Input Channels	20	A6	Analog Input Channels
8	A7		21	A8	
9	A9		22	A10	
10	A11		23	A12	
11	A13		24	A14	
12	A15		25	A16	
13	AltRef	Not Used	26	NC	Not Used

DB26 Neural Input - Differential Mode



> **Note**
>
> There are 8 (+) channels and 8 (-) channels per DB26 connector.

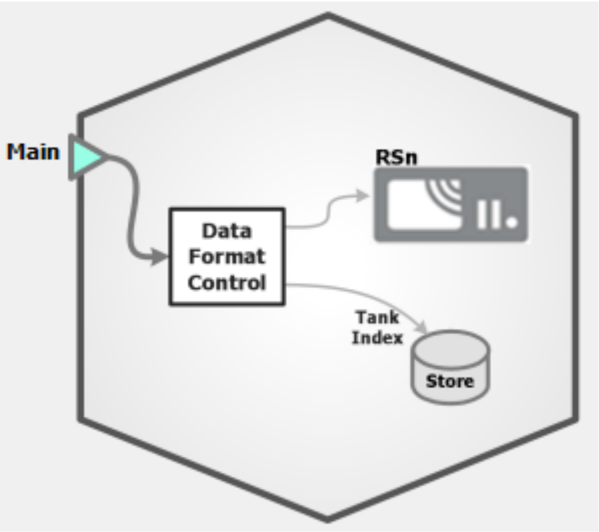
Pin	Name	Description	Pin	Name	Description
1	A1(+)	Analog Input Channels	14	V+	Positive Voltage (+2.5 V)
2	A1(-)		15	GND	Ground
3	A2(+)		16	GND	Ground
4	A2(-)		17	V-	Negative Voltage (-2.5 V)
5	NC	Not Used	18	HSD	Headstage Detect
6	HSD	Headstage Detect	19	HSD	
7	A3(+)	Analog Input Channels	20	A3(-)	Differential Input Channels
8	A4(+)		21	A4(-)	
9	A5(+)		22	A5(-)	
10	A6(+)		23	A6(-)	
11	A7(+)		24	A7(-)	
12	A8(+)		25	A8(-)	
13	GND	Ground	26	NC	Not Used

Note

Consult tech note 0896 before attempting to make any custom connections. Use the PZ5 instructions.

Data Streamers

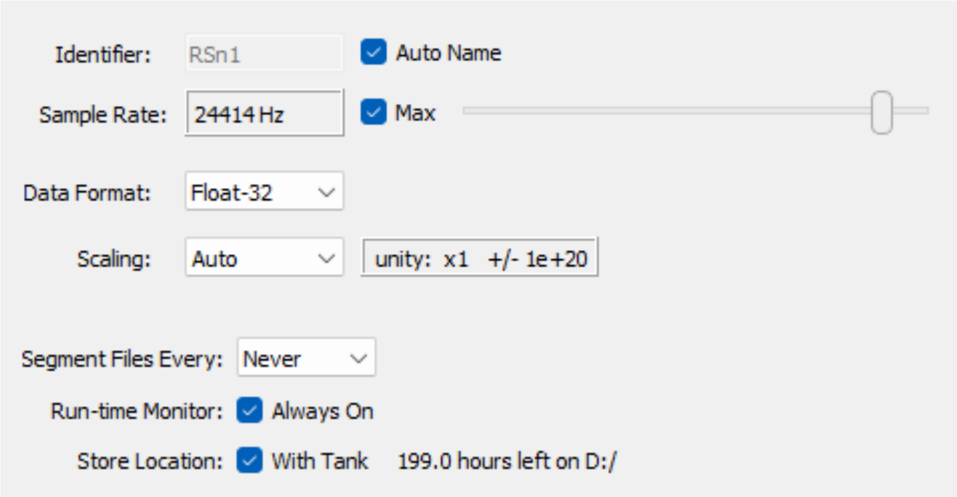
RS3 Streamer



RS3 Block Diagram

The RS3 is a DSP card for RZ2 processors with a USB3 port for streaming high bandwidth data directly back to the computer running Synapse.

RS3 Options



RS3 Options

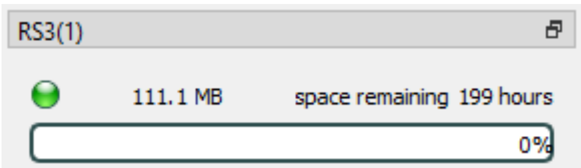
By default, the RS3 uses the RZ2 device sampling rate. You can set a rate manually by clearing the **Sample Rate Max** check box to enable the slider. You can also select the data format and any scaling. If streaming from an analog PZ5 or PZA amplifier, the Float-32 format is recommended due to the large input range and high fidelity. If using a ZD digital headstage or PZ2 amplifier, Integer-16 format with Scaling set to 1e6 is preferred.

Segment Files Every lets you split up the data files automatically at the specified interval. The new file will begin exactly one sample after the previous file. This is useful for long recordings where the user wants access to the raw data for parallel processing before the recording is over.

Run-time Monitor adds a runtime interface that warns you of data errors and USB connection issues, and shows data storage rates.

By default the data is stored in the same block folder as the rest of the recording. Use the **Store Location** option to point the RS3 data files to a different path.

RS3 Runtime Interface



RS3 Runtime Interface

The runtime interface shows the current status of the USB connection. Any errors will turn the LED red and display a message in this box.

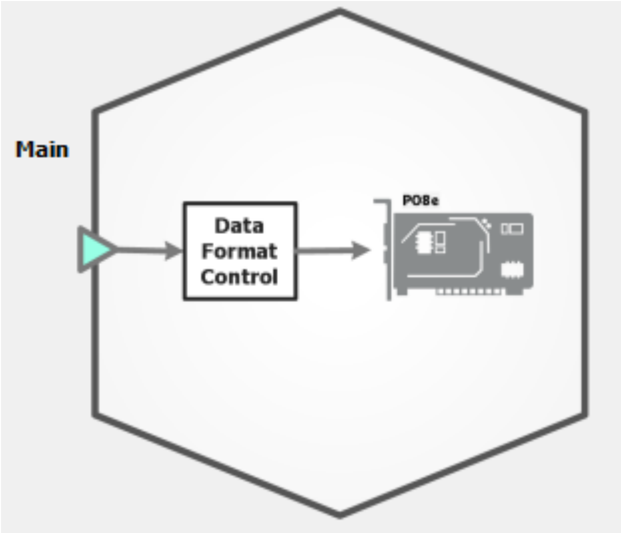
The progress bar indicates how much of the RS3 buffer is full of data that hasn't been transferred yet. This will typically be at 0% during runtime, but may spike at the beginning of the recording when the storage files are being generated, or if there are any data transfer issues to the PC.

The runtime interface also shows how much data has been recorded from the RS3 card, and an estimate of the recording time available on the hard drive.

Note

See the RS3 section of the System 3 Manual for more information about Data Streamer operations.

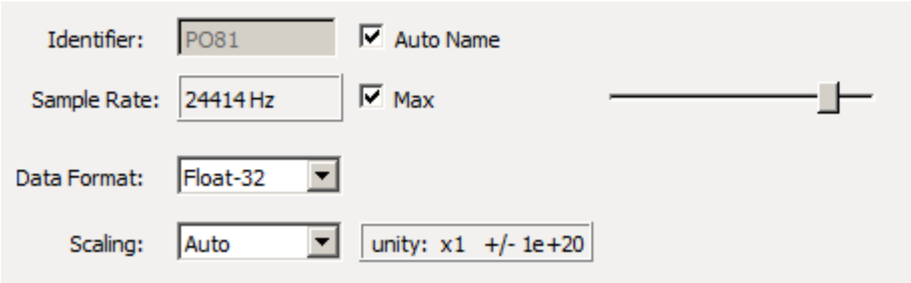
PO8e Streamer



PO8e Block Diagram

The PO8e is a data streaming device and requires a multi-channel data source, which you can select from the drop-down menu in the block diagram or drag the PO8e in the Processing Tree and drop it on the source gizmo. You can also choose the range of input channels.

PO8e Options



PO8e Options

By default, the PO8e uses the RZ device sampling rate. You can set a rate manually by clearing the **Sample Rate Max** check box to enable the slider. You can also select the data format and any scaling. If streaming from a PZ5, due to the large input range, the Float-32 format is recommended.

> **Note**
>
> See the PO8e section of the System 3 Manual for more information about PO8e Interface operations.

RZ-UDP Interface

The UDP interface can send and/or receive single or multi-channel data UDP packets from the Ethernet port labeled "UDP" on the back of the physical RZ device.

In the Rig and in the Processing Tree, the UDP functionality is split into two device objects, one for sending data (UDPSend) and one for receiving data (UDPRecv).

See the RZUDP Programming Guide for information on communicating with the RZ-UDP interface from custom applications like MATLAB and Python.

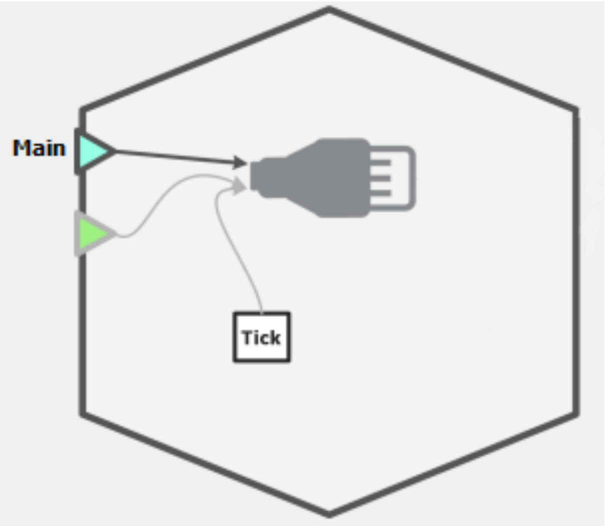
Adding UDP to your Rig and Processing Tree

The UDPSend and UDPRecv aren't added to the rig automatically.

To add them:

1. Click **Menu** then **Edit Rig**.
2. In the Rig Editor, right-click your system's RZ processor, then click **Add UDPSend** or **Add UDPRecv**.
3. Click **OK** to close the Rig Editor and update the Processing Tree.

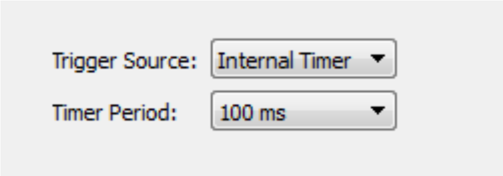
UDPSend



UDPSend Block Diagram

UDPSend sends UDP packets out the RZ to external devices. You can select the source and range of channels from the block diagram drop-down menu, or drag the UDPSend icon in the Processing Tree and drop it on a source gizmo.

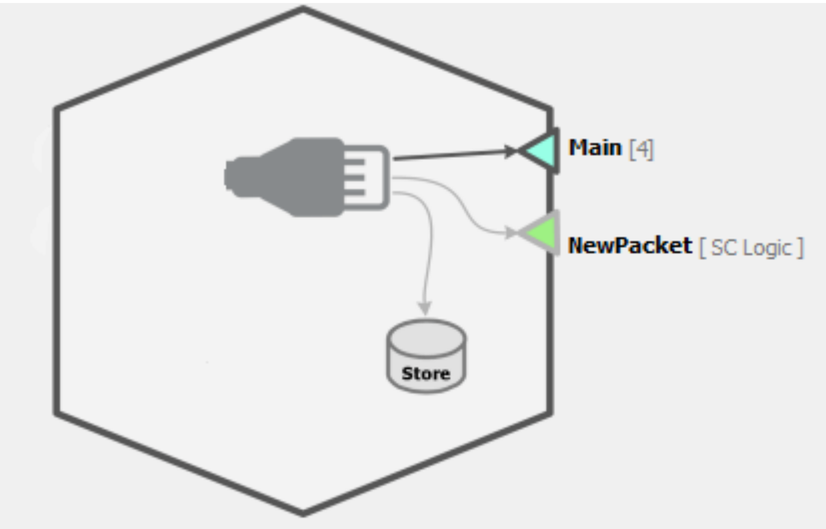
UDPSend Options



Send to UDP Options

You can set the time period for triggering and choose either an internal real-time clock, or a secondary gizmo input as the trigger source.

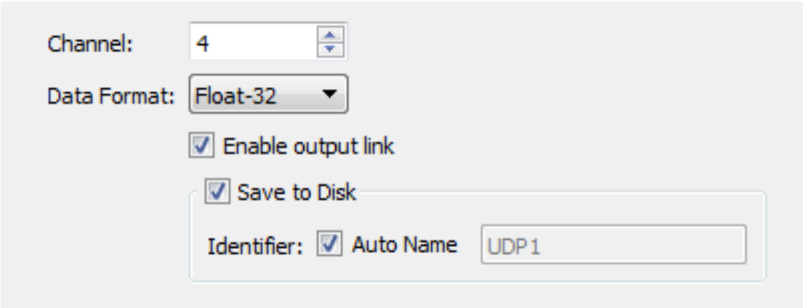
UDPRecv



UDPRecv Block diagram

UDPRecv receives data packets through the UDP interface from an external device and makes them available for further real-time processing.

UDPRecv Options



Receive from UDP Options

You can choose the number of expected channels in the incoming UDP packets and select the data type from a drop down list, so that the RZ device knows how to convert the received bits into the correct data source output.

You can select the **Enable output link** check box to make the received packet and timing signal data source outputs available to use with other gizmos.

Select the **Save to Disk** check box to store the received packet in the data tank as a timestamped event.

See the UDP section of the System 3 Manual for more information about UDP operations.

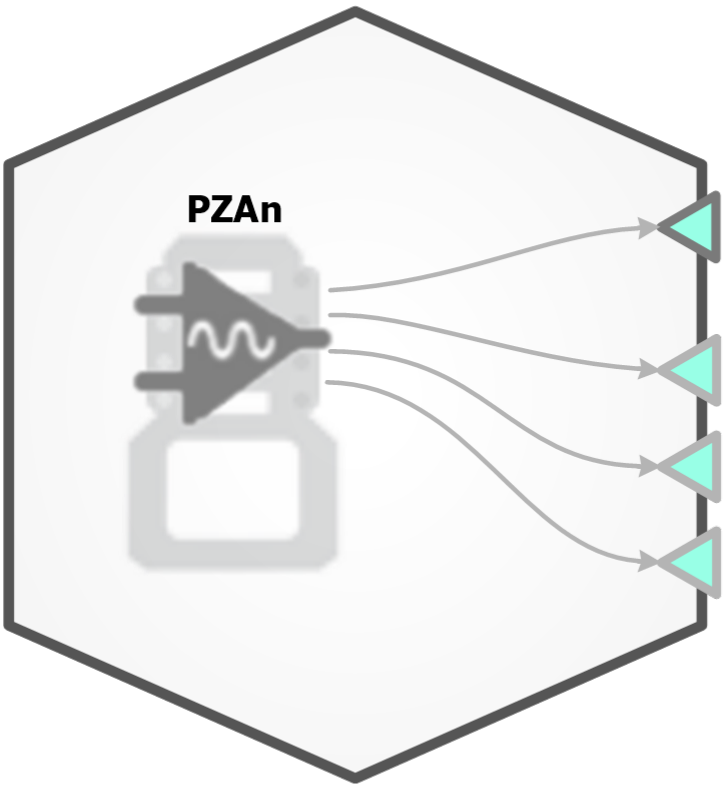
Subject Interface Amplifiers

The Subject Interface Module (SIM) is the control object for the SI2, SI4, and SI8 devices. It connects to a DSP-M or optical QDSP card in your RZ processor. The Subject Interface is configurable with stimulator cards (IZV), analog amplifier cards (PZA), or digital headstage interface cards (PZD). Because of the diversity in functionality of this device, each card type has its own "Sub-HAL" object within Synapse that is used to configure those cards. The acquisition and stimulation all happen within the Sub-HALs, which are all independent objects in the Processing Tree.

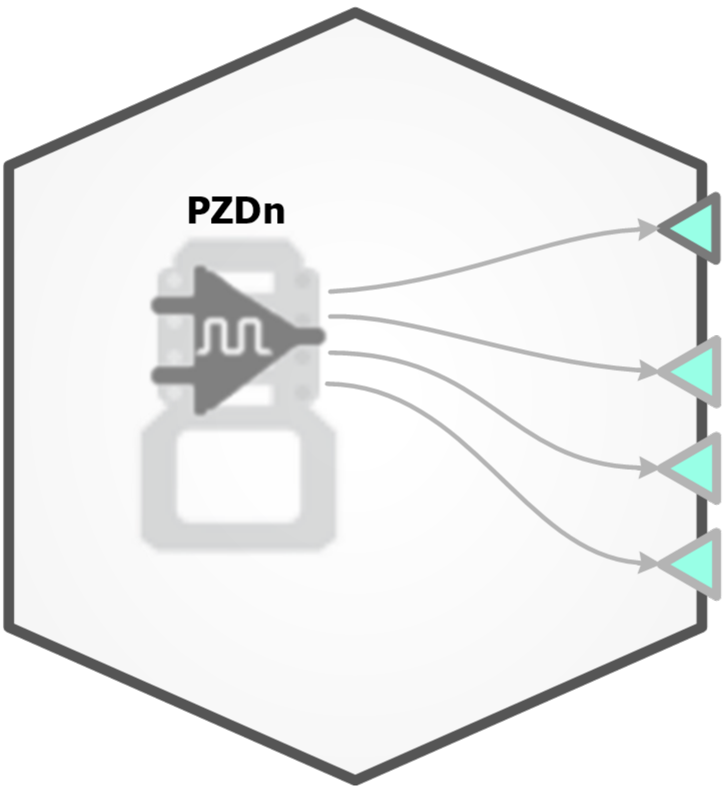
PZA Analog Amplifier & PZD Digital Headstage Interface

The PZA & PZD can split the available boards into sub-amps which are completely isolated from one another.

PZA Block Diagram



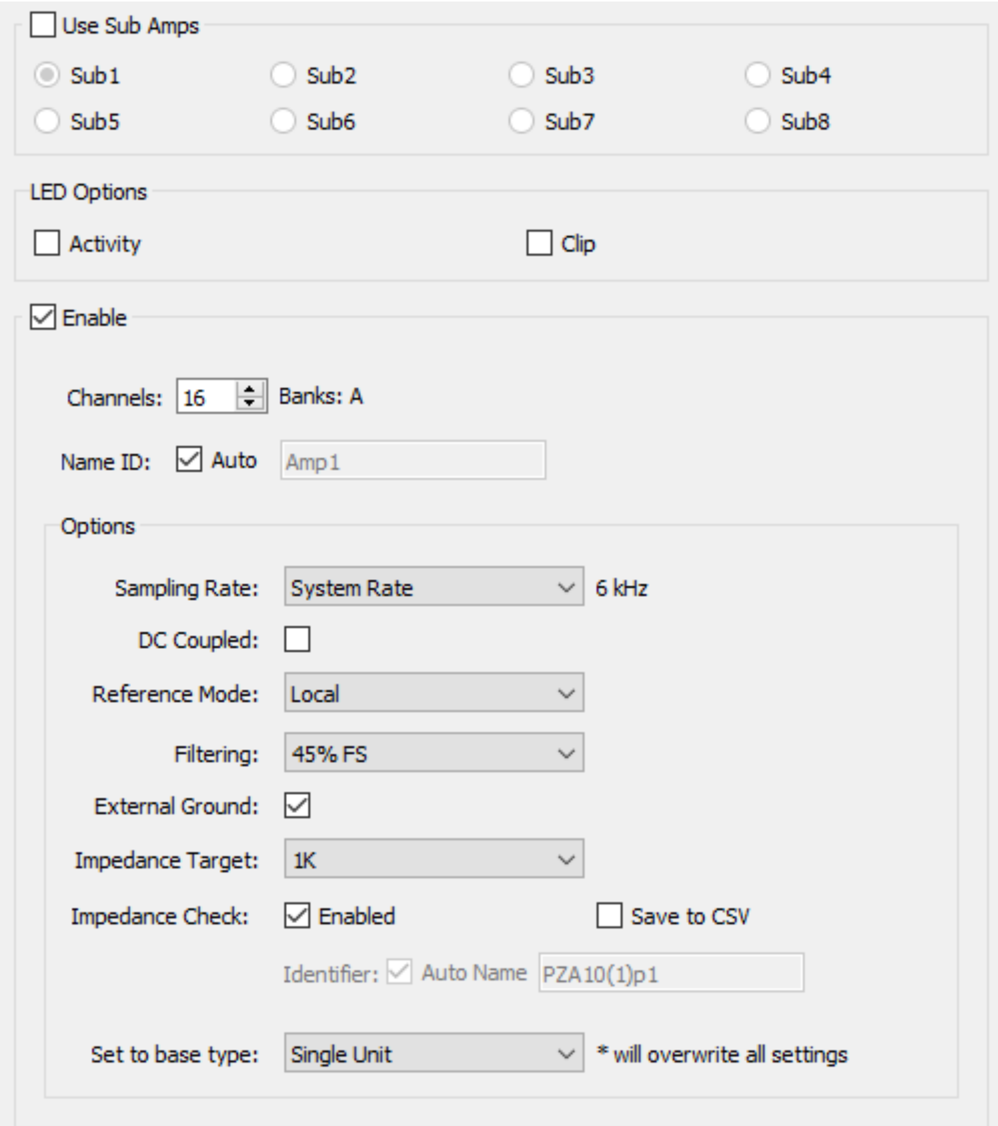
PZD Block Diagram



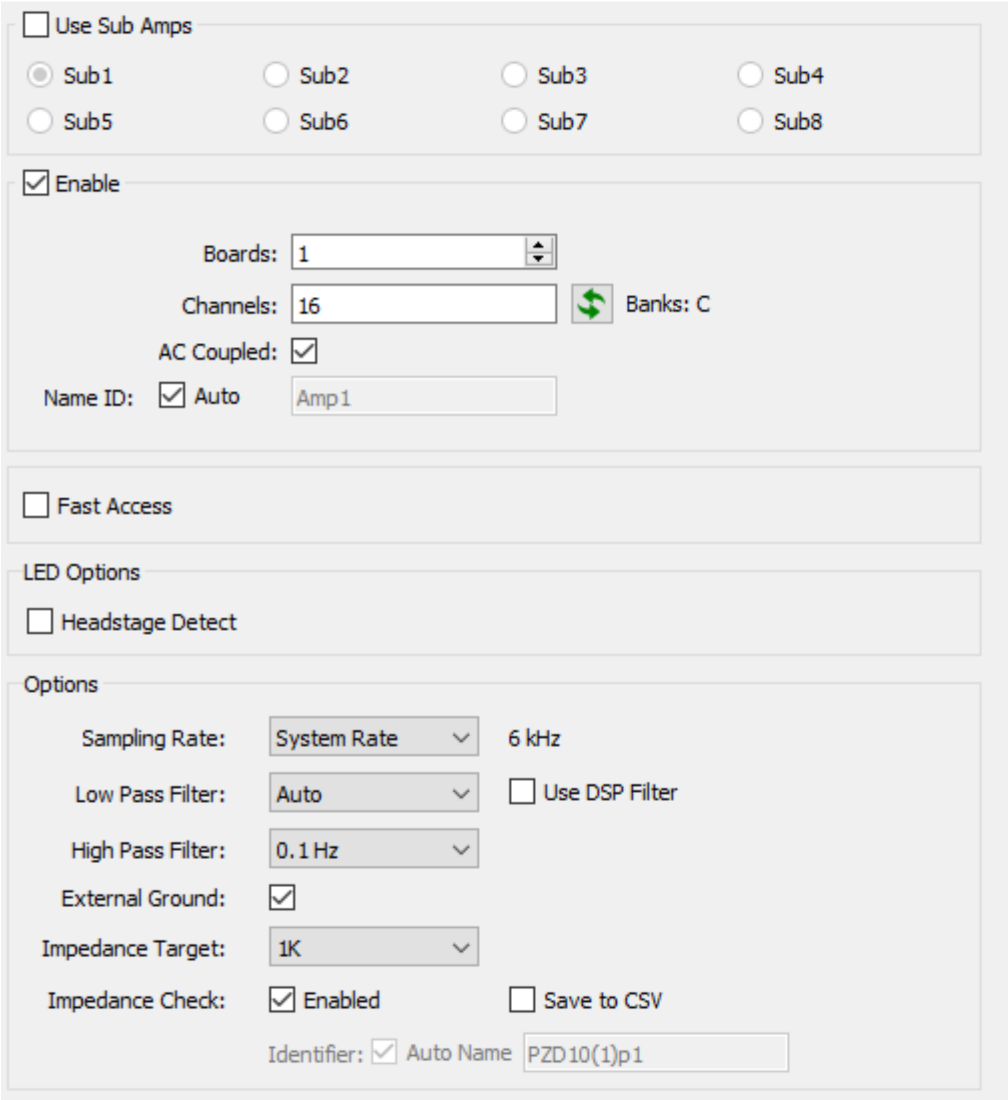
PZA & PZD Options

PZA Analog Options and PZD Digital Options.

PZA Analog Options



PZD Digital Options



PZA Analog Option Descriptions

Option	Description
Use Sub Amps	Divide the input channels into "logical amplifiers" that can record different types of signals, at different rates and referencing modes with independent grounds. All of the below configuration options apply only to the selected sub amp.
Boards	Specify the number of boards to include in this sub-amp. Selectable values are limited by the Rig configuration.
Channels	The channel count must be at least four and must be a multiple of 2. The corresponding physical banks on the SI are displayed to the right. In Differential Reference Mode set the total electrode count here. The channels from the PZA sub-amp are mapped for you to remove duplicates on the output link.
Name ID	Choose the name for this sub amp data source that will be visible to other gizmos
LED Activity	Tell the PZA whether to flash the green activity LEDs on its front panel when activity is detected
LED Clipping	Tell the PZA whether to flash the red clipping LEDs when the signal is close to saturating the amplifier
Sampling Rate	Set the Sampling Rate to match the desired frequency band of your incoming signals (or leave at 'System Rate' if you are unsure). By default, the sampling rate matches that of the RZ.
DC Coupled	Remove the 0.4 Hz high pass filter on the input signals and record DC potentials.
Reference Mode	Select the mode from the drop-down menu. Local - each bank of 16 uses its own reference (pin 5 on the DB26). Shared - all channels in sub-amp share the same reference (pin 5 of first bank in sub-amp). None - the ground connection is used as the reference. Differential - each even channel acts as a reference for the odd channel before it. Note: the output channels will be mapped for you to remove duplicate channels.
Filtering	Set the anti-aliasing low pass filter cutoff as a percentage of the sampling rate
External Ground	Connect this sub-amp ground to the external ground plug on the physical SIM device. **Caution:** When using multiple sub-amps make sure they aren't all sharing the External Ground connection or else they won't be isolated!
Impedance Check	Enable run-time impedance check
Save to CSV	Log impedance values into CSV file stored within the block folder whenever an impedance check runs
Set to base type	If you are unsure, use Set to Base Type to configure the amp with default settings based on common signal types

PZD Digital Analog Option Descriptions

Option	Description
Use Sub Amps	Divide the input channels into "logical amplifiers" that can record different types of signals, at different rates and referencing modes with independent grounds. All of the below configuration options apply only to the selected sub amp.
Boards	Specify the number of boards to include in this sub-amp. Selectable values are limited by the Rig configuration.
Channels	The channel count must be at least four and must be a multiple of 2. The corresponding physical banks on the SI are displayed to the right. Click the Refresh button to automatically detect the connected headstage count (your RZ hardware must be turned on and connected for this to work).
AC Coupled	Apply an additional 0.4 Hz high pass filter to the incoming signals.
Name ID	Choose the name for this sub amp data source that will be visible to other gizmos
Fast Access	Select this option to read the data as 16-bit to reduce cycle usage on the DSP. Required for acquiring 256 channels of digital headstage inputs.
Headstage Detect	Tell the PZD whether to indicate the headstage channel count via the front panel LEDs
Sampling Rate	Set the Sampling Rate to match the desired frequency band of your incoming signals (or leave at 'System Rate' if you are unsure). By default, the sampling rate matches that of the RZ.
Low Pass Filter	Select a cutoff frequency for a lowpass filter that is implemented on the digital headstage. Set to 'Auto' to match it to the PZD sub-amp sampling rate. See table below.
Use DSP Filter	Add an additional lowpass digital filter implemented on the RZ, set to the same frequency as the Low Pass Filter setting above, to remove high frequency digital noise from the incoming signal that was added by the digitizing chip
High Pass Filter	Select a cutoff frequency for a highpass filter that is implemented on the digital headstage
External Ground	Connect this sub-amp ground to the external ground plug on the physical SIM device. **Caution:** When using multiple sub-amps make sure they aren't all sharing the External Ground connection or else they won't be isolated!
Impedance Target	Use to intelligently control the impedance checking circuit on the headstage based on your target impedance
Impedance Check	Enable run-time impedance check
Save to CSV	Log impedance values into CSV file stored within the block folder whenever an impedance check runs

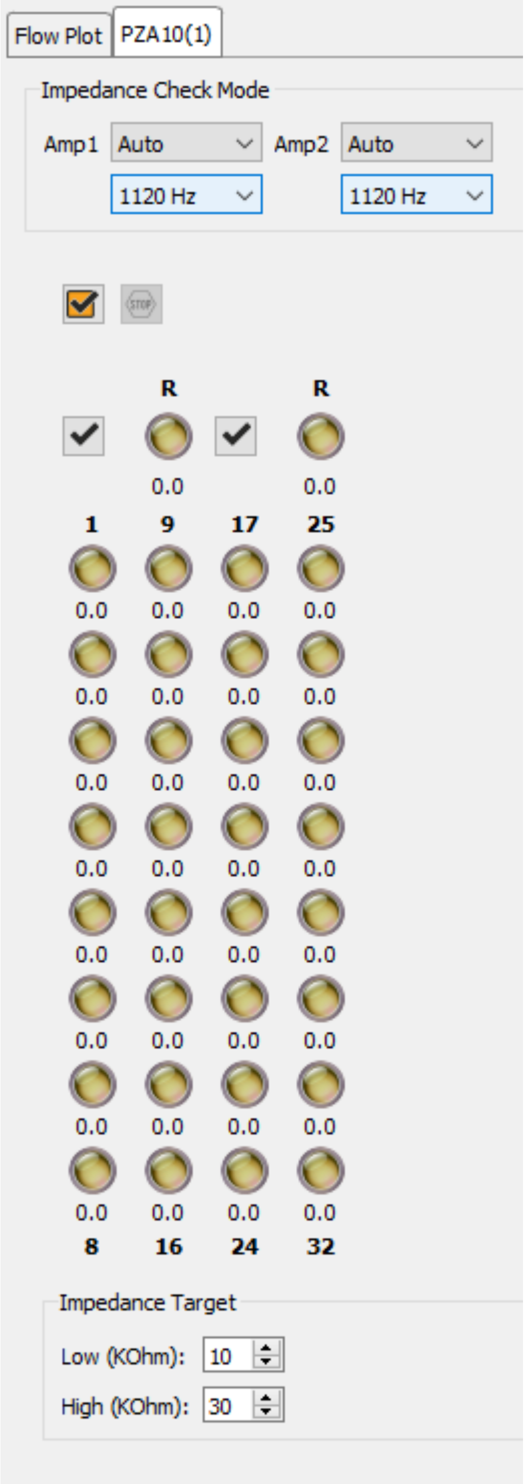
'Auto' low pass filter settings for different sub amp sampling rates

Sampling Rate	LP Auto Filter
750 Hz	300 Hz
1.5 kHz	750 Hz
3 kHz	1.5 kHz
6 kHz	3 kHz
12 kHz	5 kHz
25 kHz	10 kHz

See the SIM section of the System 3 Manual for more information about configuring and using the PZA amplifier and PZD interface.

Runtime Interface

When **Impedance Check** is checked, a user interface appears at runtime.



The PZA/PZD Tab provides an interface for impedance checking on all channels. The display represents the stimulation channels divided into banks of eight channels, broken up by sub-amplifier.

The dropdowns at the top define which type of inputs to check and the probe frequency. If a reference electrode is used in the PZA subamp configuration, an LED for the reference is shown with that subamp and it is an available test option. If there are more than one type of input, 'Auto' loops through all of them during the impedance check.

Running the Impedance Check

Run an impedance check on this sub-amp. The PZA/PZD enters impedance check mode and the values are displayed on the interface.

Check All. Run an impedance check on all sub-amps as described above.

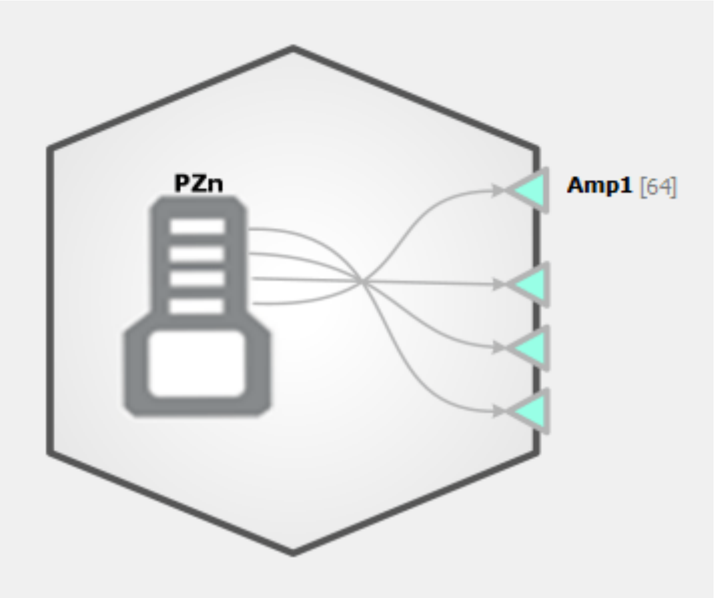
Stop Checking. Stops the impedance checking prematurely.

Results of impedance check are indicated by color: below low impedance threshold (green), above high impedance threshold (red), between low and high impedance thresholds (yellow). The actual impedance values (in kOhm) are displayed beneath each indicator.

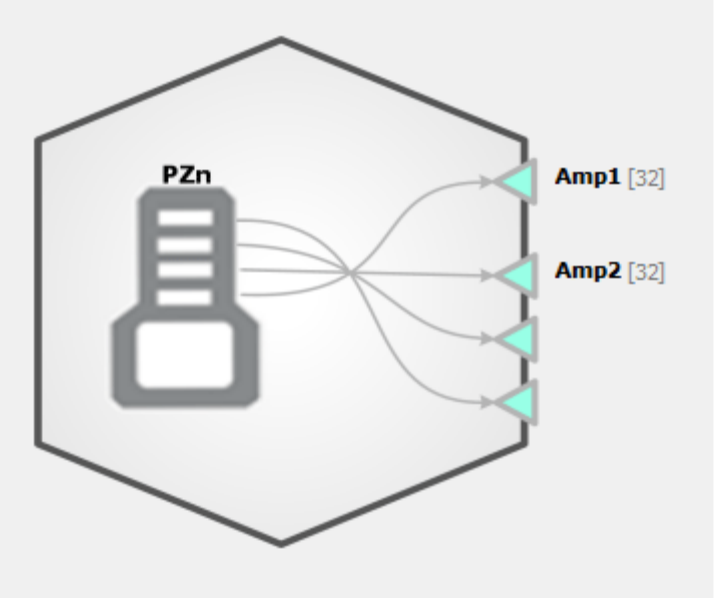
PZ Amplifiers

PZ Block Diagrams, showing a single amp (left) and sub amps (right).

PZ5 Single Amp



PZ5 Sub Amps

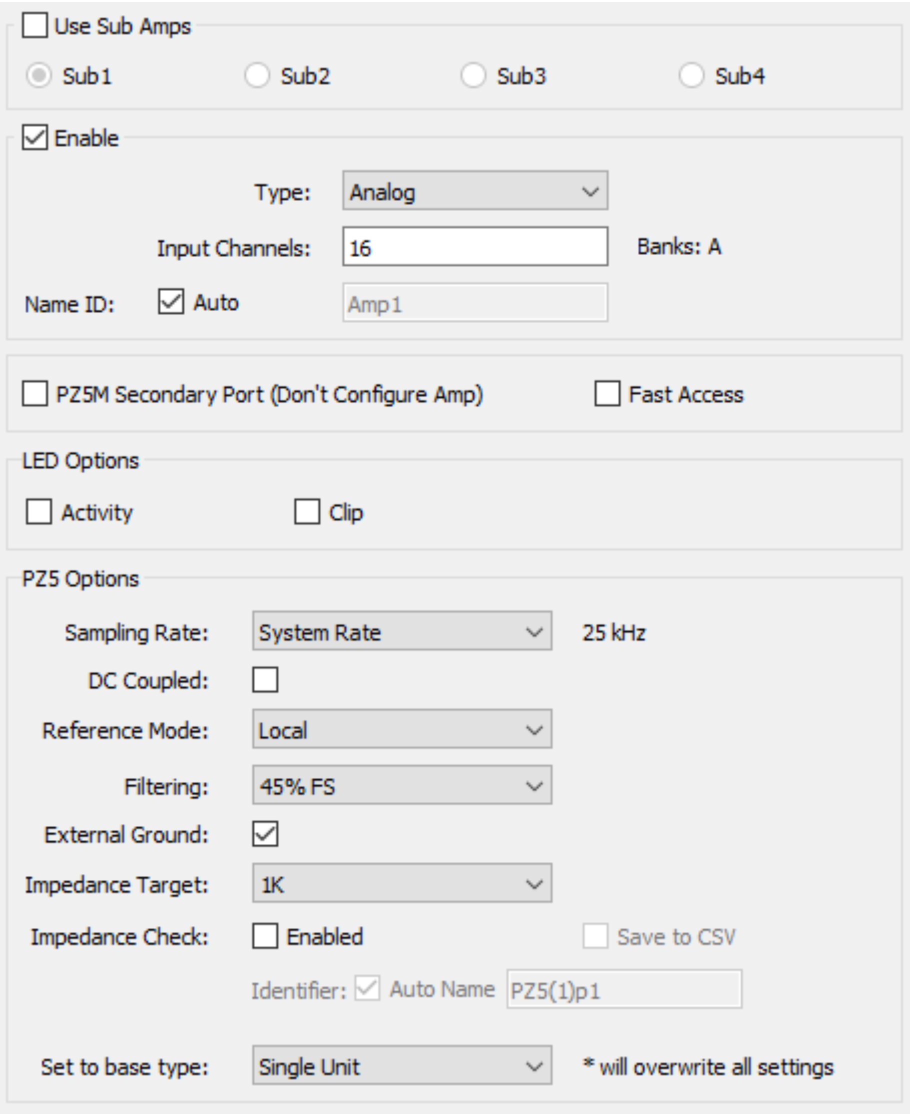


The PZ amplifier HAL reads data from a connected PZn amplifier. The options vary for each amplifier model. The PZ5 amplifier can have up to four logical sub amps; each sub amp is individually configurable and forms a multi-channel floating point data sources that can be linked to other gizmos. All other PZs have one multi-channel floating point data source.

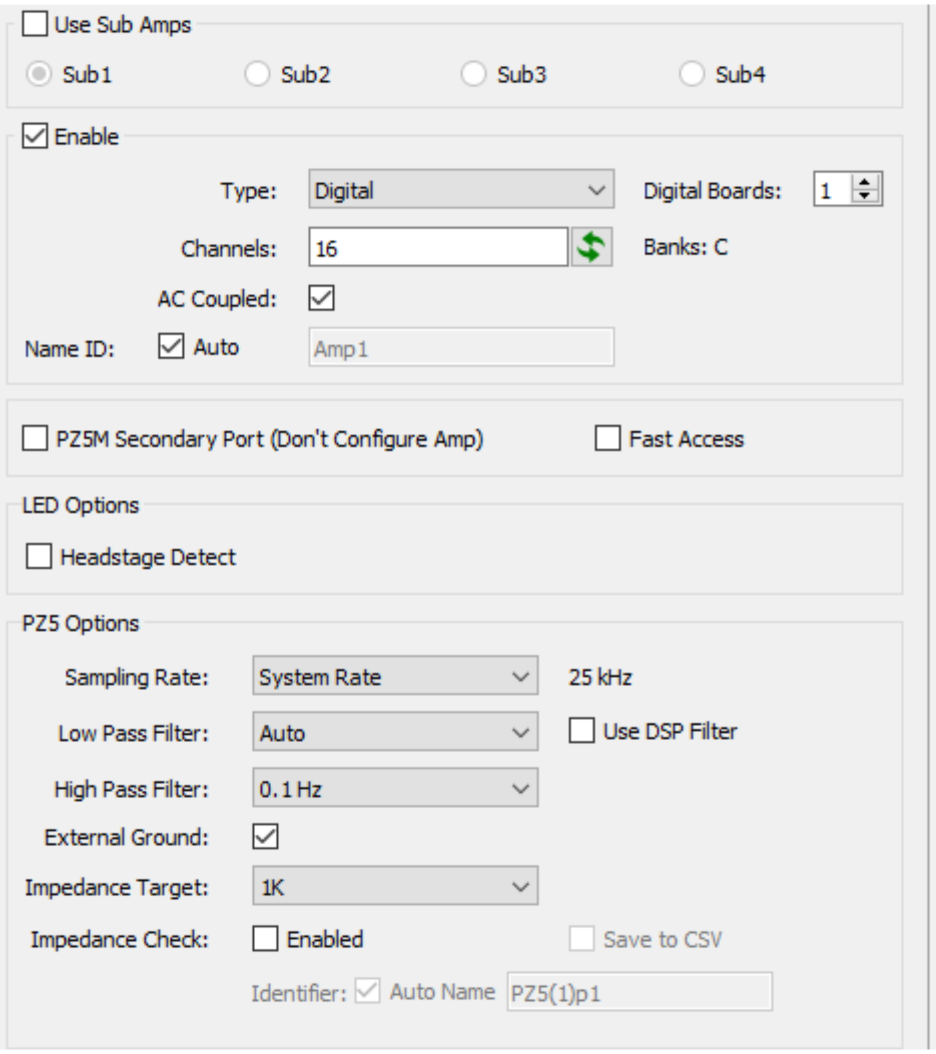
PZ5 Options

PZ5 Analog and Digital Options.

PZ5 Analog Options



PZ5 Digital Options



PZ5 Analog Option Descriptions

Option	Description
Use Sub Amps check box	Select to divide input channels into logical sub amps that can be used to record different types of signals, at different rates, referencing modes, and other settings.
Sub Amps radio buttons	Select a radio button to view and edit settings for the corresponding sub amp: Sub1, Sub2, Sub3, or Sub4. When sub amps are used, all of the below configuration options apply only to the selected sub amp. Each sub amp that will be used must be configured separately. If a conflict or error is detected as a result of any changed settings, Synapse displays the relevant sub amp settings and a red warning.
Enable	Select to enable the selected sub amp.
Type	Use an Analog or Digital amp board. When using Digital Amps, specify the number of boards. Selectable values are limited by the Rig configuration.
Channels	Type or click arrow keys to set the number of channels. The channel count must be at least four and must be a multiple of 2. The corresponding physical bank of channels on the PZ5 is displayed to the right. Note: In Differential Reference Mode, the channels from the PZ5 sub-amp are mapped for you to remove duplicates.
Name ID	Choose the name for this sub amp data source that will be visible to other gizmos.

Option	Description
PZ5M Secondary Port (Don't Configure Amp)	When using a PZ5M with either 256 or 512 channels, two PZ5 HALs may be specified in the Rig Editor, one for each fiber optic connection from the PZ5M. Only the fiber connected to the Primary port can configure the PZ5M. Use this check box on the PZ5 HAL connected to the Secondary port on the PZ5M. This setting disables the HAL configuration options and only reads the channel data from the port.
Fast Access	For a PZn connected to a DSPP card, select to perform 16-bit data reads to reduce cycle usage on the DSP.
LED Activity	Tell the PZ5 whether to flash the green activity LEDs on its front panel when activity is detected
LED Clipping	Tell the PZ5 whether to flash the red clipping LEDs when the signal is close to saturating the amplifier

Option	Description
Sampling Rate	Set the Sampling Rate to match the desired frequency band of your incoming signals (or leave at 'System Rate' if you are unsure). By default, the sampling rate matches that of the RZ.
External Ground	Connect this sub-amp ground to the external ground plug on the physical PZ5 device. **Caution:** When using multiple sub-amps make sure they aren't all sharing the External Ground connection or else they won't be isolated!
DC Coupled	Remove the 0.4 Hz high pass filter on the input signals and record DC potentials.
Reference Mode	Select the mode from the drop-down menu. Local - each bank of 16 uses its own reference (pin 5 on the DB26). Shared - all channels in sub-amp share the same reference (pin 5 of first bank in sub-amp). None - the ground connection is used as the reference. Differential - each even channel acts as a reference for the odd channel before it. Note: the output channels will be mapped for you to remove duplicate channels.
Filtering	Set the anti-aliasing low pass filter cutoff as a percentage of the sampling rate.
Impedance Target	This only affects the displayed impedance text colors on the touchscreen when running an impedance check. Must be using a passive headstage to run impedance check.
Impedance Check	Enable run-time impedance check
Save to CSV	Log impedance values into CSV file stored within the block folder whenever an impedance check runs
Set to base type	If you are unsure, use Set to Base Type to configure the amp with default settings based on common signal types.

PZ5 Digital Option Descriptions

Option	Description
Use Sub Amps check box	Select to divide input channels into logical sub amps that can be used to record different types of signals, at different rates, filters, and other settings
Sub Amps radio buttons	Select a radio button to view and edit settings for the corresponding sub amp: Sub1, Sub2, Sub3, or Sub4. When sub amps are used, all of the below configuration options apply only to the selected sub amp. Each sub amp that will be used must be configured separately. If a conflict or error is detected as a result of any changed settings, Synapse displays the relevant sub amp settings and a red warning.
Enable	Select to enable the selected sub amp
Type	Use an Analog or Digital amp board. When using Digital Amps, specify the number of boards. Selectable values are limited by the Rig configuration.
Channels	The channel count must be at least four and must be a multiple of 2. The corresponding physical banks on the PZ5 are displayed to the right. Click the Refresh button to automatically detect the connected headstage count (your RZ hardware must be turned on and connected for this to work).
Name ID	Choose the name for this sub amp data source that will be visible to other gizmos

Option	Description
PZ5M Secondary Port (Don't Configure Amp)	When using a PZ5M with either 256 or 512 channels, two PZ5 HALs may be specified in the Rig Editor, one for each fiber optic connection from the PZ5M. Only the fiber connected to the Primary port can configure the PZ5M. Use this check box on the PZ5 HAL connected to the Secondary port on the PZ5M. This setting disables the HAL configuration options and only reads the channel data from the port.
Fast Access	For a PZn connected to a DSPP card, select to perform 16-bit data reads to reduce cycle usage on the DSP
Headstage Detect	Tell the PZ5 whether to indicate the headstage channel count via the front panel LEDs

Option	Description
Sampling Rate	Set the Sampling Rate to match the desired frequency band of your incoming signals (or leave at 'System Rate' if you are unsure). By default, the sampling rate matches that of the RZ.
External Ground	Connect this sub-amp ground to the external ground plug on the physical PZ5 device. **Caution:** When using multiple sub-amps make sure they aren't all sharing the External Ground connection or else they won't be isolated!
AC Coupled	Apply a 0.4 Hz high pass filter to the incoming signals
Low Pass Filter	Select a cutoff frequency for a lowpass filter that is implemented on the digital headstage. Set to 'Auto' to match it to the sub-amp sampling rate. See table below.
Use DSP Filter check box	Add an additional lowpass digital filter implemented on the RZ, set to the same frequency as the Low Pass Filter setting above, to remove high frequency digital noise from the incoming signal that was added by the digitizing chip.
High Pass Filter	Select a cutoff frequency for a highpass filter that is implemented on the digital headstage
Impedance Target	Use to intelligently control the impedance checking circuit on the headstage based on your target impedance
Impedance Check	Enable run-time impedance check
Save to CSV	Log impedance values into CSV file stored within the block folder whenever an impedance check runs

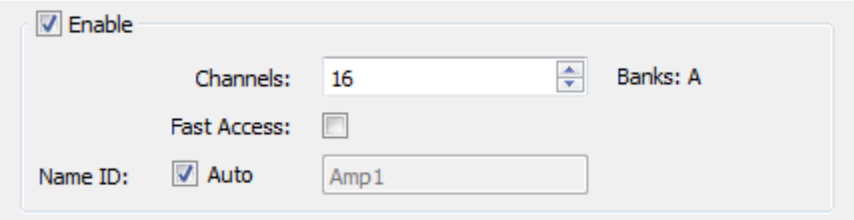
'Auto' low pass filter settings for different sub amp sampling rates

Sampling Rate	LP Auto Filter
750 Hz	300 Hz
1.5 kHz	750 Hz
3 kHz	1.5 kHz
6 kHz	3 kHz
12 kHz	5 kHz
25 kHz	10 kHz

See the PZ5 section of the System 3 Manual for more information about configuring and using the PZ5 amplifier.

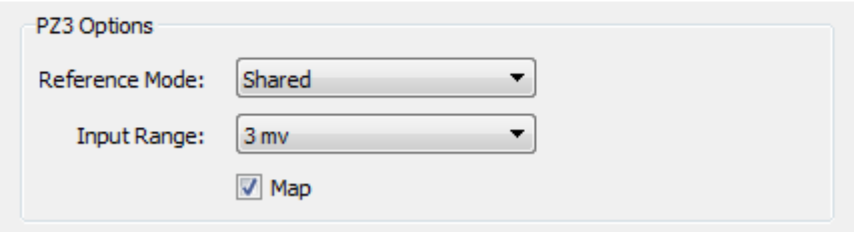
PZ2 / PZ3/ PZ4 Options

PZ2 Options



Option	Description
Channels	Type or click arrow keys to set the number of channels.
Fast Access	For a PZ2 connected to a DSPP card, select to perform 16-bit data reads to reduce cycle usage on the DSP.
Name ID	Choose the name for this data source that will be visible to other gizmos.

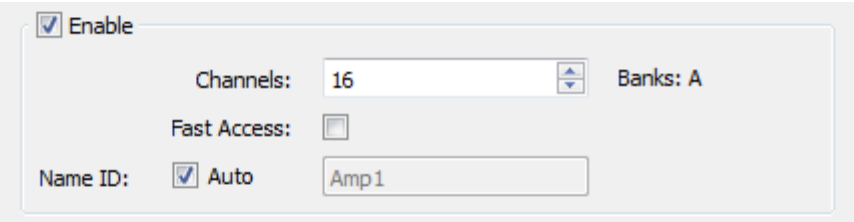
PZ3 Options



Option	Description
Channels	Type or click arrow keys to set the number of channels.
Fast Access	For a PZ3 connected to a DSPP card, select to perform 16-bit data reads to reduce cycle usage on the DSP.
Name ID	Choose the name for this data source that will be visible to other gizmos.
Reference Mode	Select the mode from the drop-down menu. Shared - all channels use a separate shared reference. Differential - each even channel acts as a reference for the odd channel before it.
Input Range	Select the desired maximum input range.
Map	(Recommended) Select to remove the even channels and use only the signal channels in the data source output.

See the PZ3 section of the System 3 Manual for more information about the operational modes of the PZ3 amplifier.

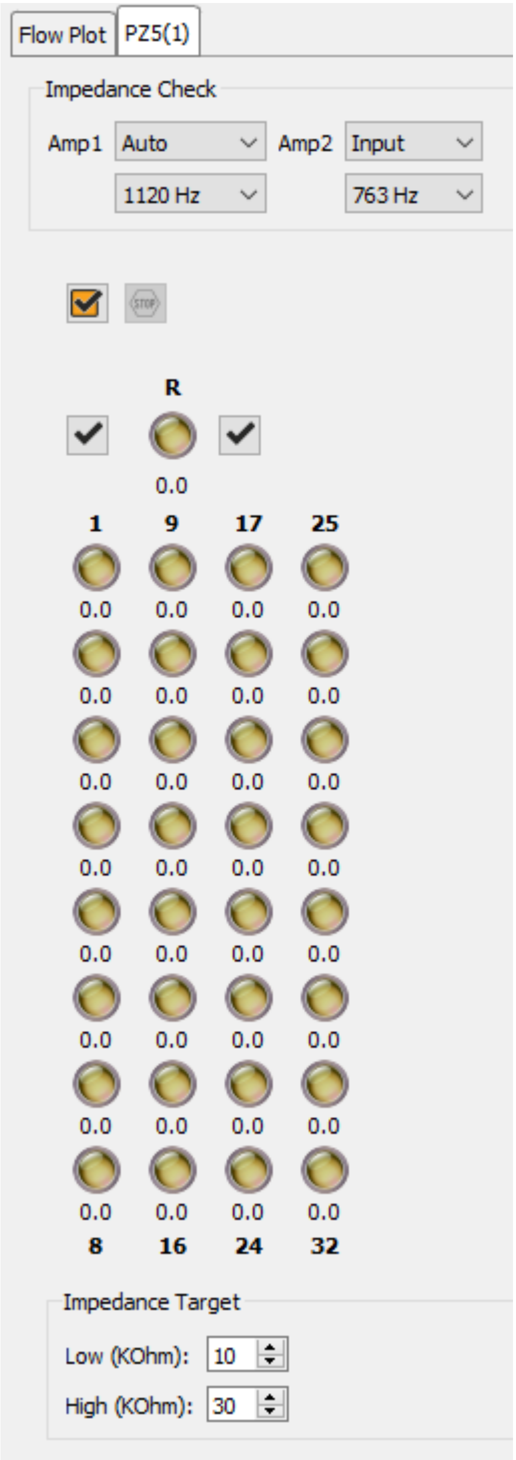
PZ4 Options



Option	Description
Channels	Type or click arrow keys to set the number of channels.
Fast Access	For a PZ4 connected to a DSPP card, select to perform 16-bit data reads to reduce cycle usage on the DSP.
Name ID	Choose the name for this data source that will be visible to other gizmos.

Runtime Interface

When **Impedance Check** is checked, a user interface appears at runtime.



The PZ5 tab provides an interface for impedance checking on all channels. The display represents the stimulation channels divided into banks of eight channels, broken up by sub-amplifier.

The dropdowns at the top define which type of inputs to check and the probe frequency. If a reference electrode is used in the subamp configuration, an LED for the reference is shown with that subamp and it is an available test option. If there are more than one type of input, 'Auto' loops through all of them during the impedance check.

Running the Impedance Check

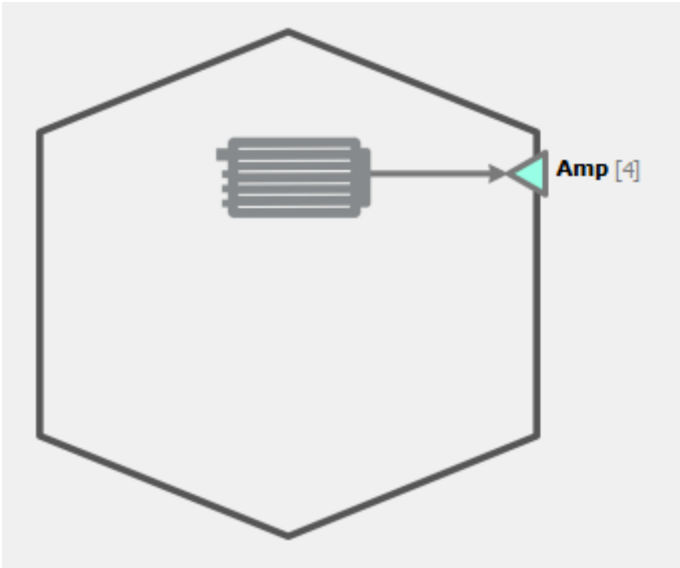
Run an impedance check on selected sub-amplifier. The PZ5 enters impedance check mode and the values are displayed on the interface.

Check All. Run an impedance check on all sub-amps as described above.

Stop Checking. Stops the impedance checking prematurely.

Results of impedance check are indicated by color: below low impedance threshold (green), above high impedance threshold (red), between low and high impedance thresholds (yellow). The actual impedance values (in kOhm) are displayed beneath each indicator.

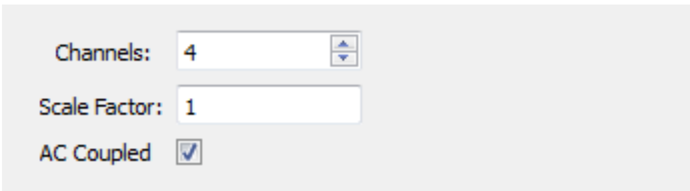
RA Amplifiers



RA Amplifier Block Diagram

The RA Hal reads data from a legacy optic port on the front panel of the RZ / RX device. Supported amplifiers include the RA4PA, RA16PA, RA8GA, and Medusa4Z.

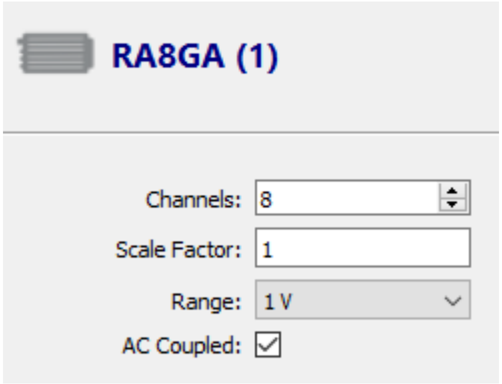
RA Amplifier Options



RA Amp Options

Select the number of channels and apply an optional scale factor to the incoming signals. If **AC Coupled** is selected, a 0.4 Hz highpass filter is applied to the incoming signals.

If using an RA8GA, the voltage range must also be set to match the voltage selected on the device front panel.



RA8GA options

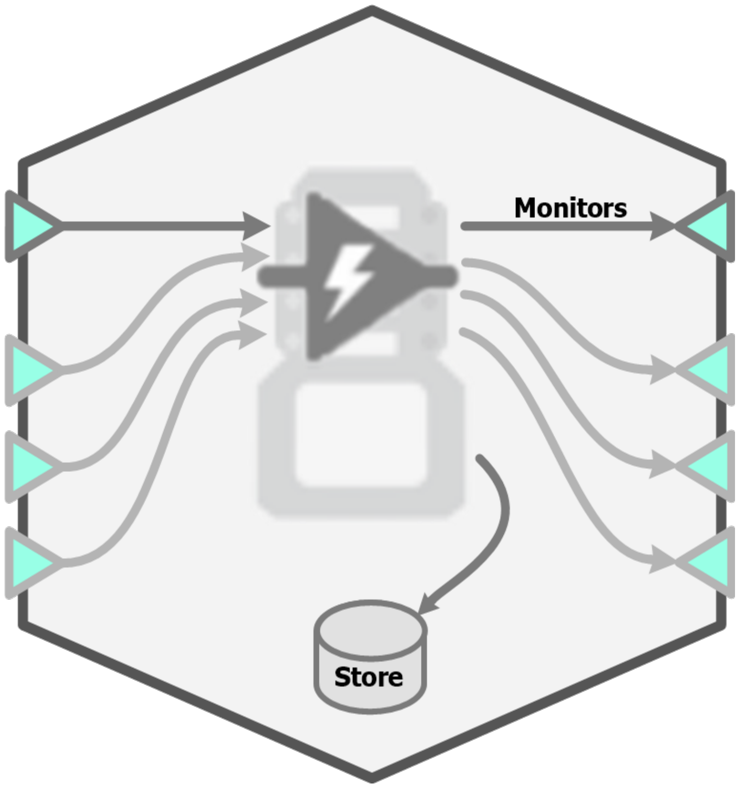
Subject Interface Stimulation

The Subject Interface Module (SIM) is the control object for the SI2, SI4, and SI8 devices. It connects to a DSP-M or optical QDSP card in your RZ processor. The Subject Interface is configurable with stimulator cards (IZV), analog amplifier cards (PZA), or digital headstage interface cards (PZD). Because of the diversity in functionality of this device, each card type has its own "Sub-HAL" object within Synapse that is used to configure those cards. The acquisition and stimulation all happen within the Sub-HALs, which are all independent objects in the Processing Tree.

> **Tip**
>
> See the Electrical Stimulation Guide for help with experiment design.

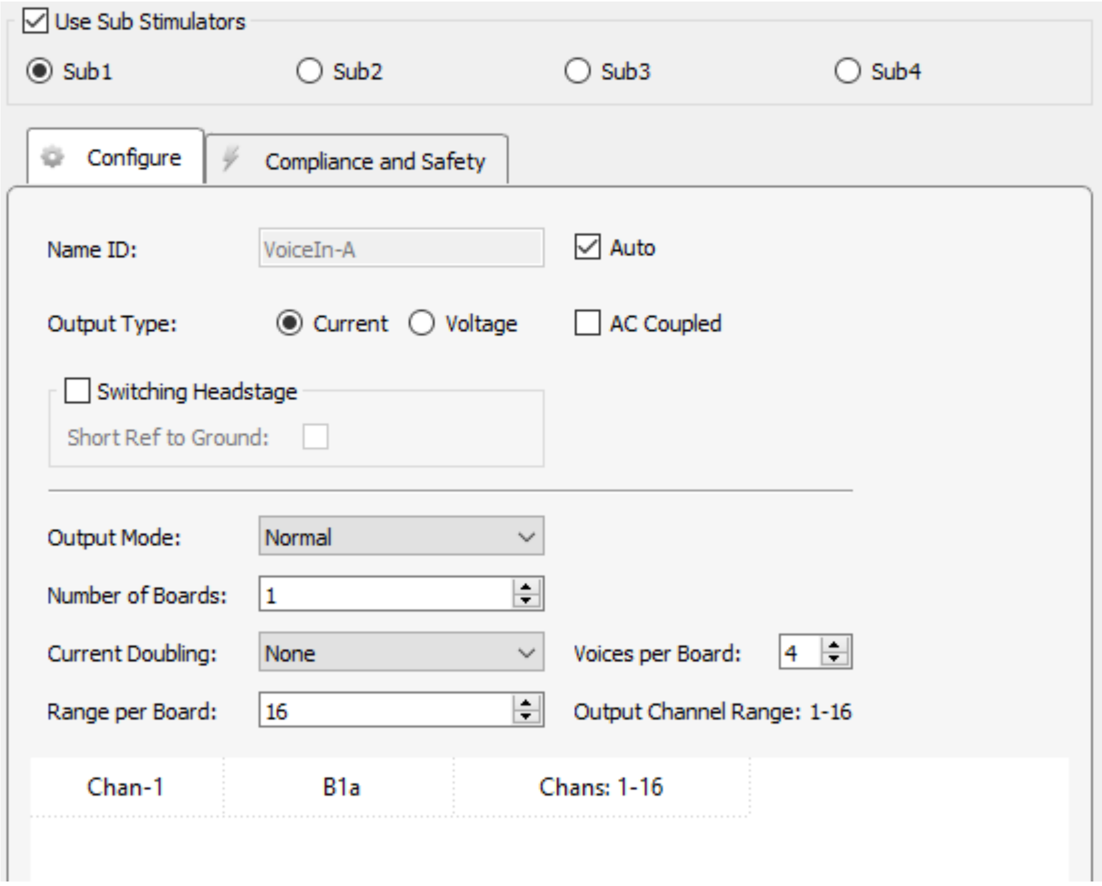
IZV Stimulator

The IZV can split the available boards into sub-stimulators which are completely isolated from one another. It is typically controlled by the Electrical Stim Driver gizmo.



IZV Block Diagram

IZV Options



IZV Stimulator Configure Tab Options

When using Sub Stimulators, each sub-stimulator must be configured independently. Each sub-stimulator has its own multi-channel input signal. For example, the first sub-stimulator uses the gizmo input called "VoiceIn-A", the second sub-stimulator uses "VoiceIn-B" gizmo input, and so on. It is generally expected that this signal comes from an Electrical Stim Driver gizmo so that it has the proper format. The format is a multi-channel signal where the odd channels contain the actual stim waveforms and the even channels contain the target stim channel. For an input that has 4 unique stim signals, it will be an 8-channel signal where channels 1, 3, 5, 7 are the stim signals and channels 2, 4, 6, 8 are the corresponding target stim channels.

Important

The sub stimulators must be ordered from highest to lowest bank count. For example, if you want a 2 bank stimulator and a 4 bank stimulator, then Sub Stimulator 1 should have 4 banks and Sub Stimulator 2 should have 2 banks.

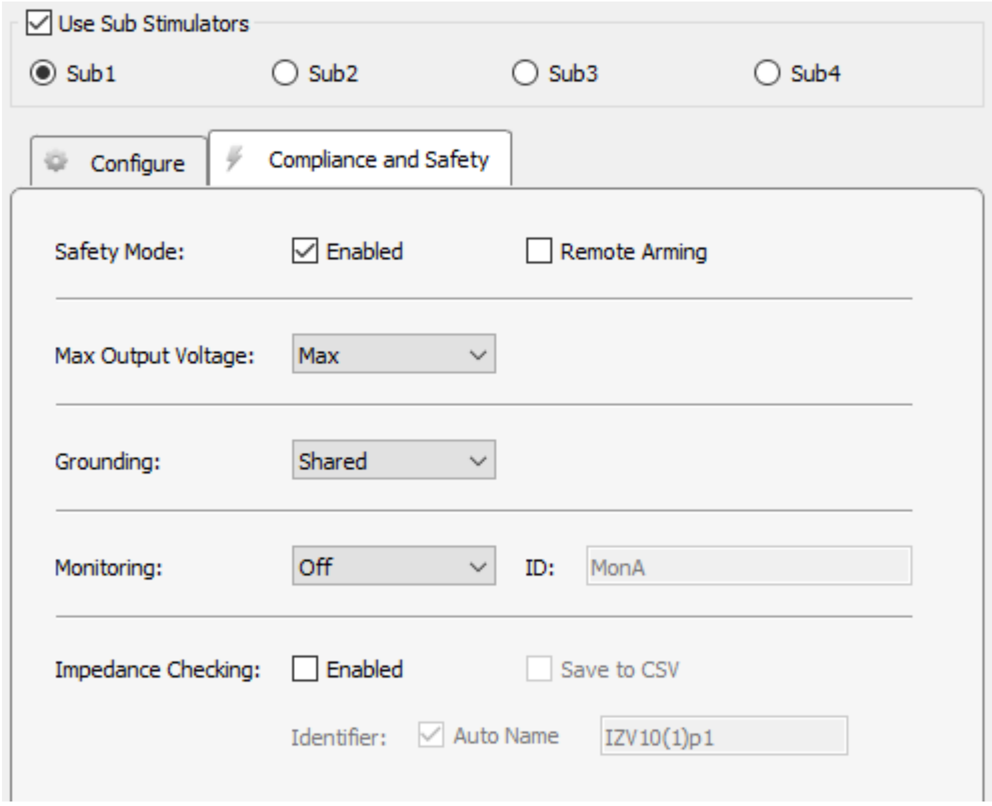
Each physical IZV board has 4 unique hardware voices that can be routed to its 16 output channels.

The data table at the bottom shows the mapping between the gizmo input voices and the physical hardware voices. The first column is the voice of the "VoiceIn" for this sub-stimulator. The middle column is the board number within the sub- stimulator that it uses and the hardware voice on that board it uses (a, b, c, d). The last column is the range of channel numbers that the VoiceIn stim channels should address.

Check the **Switching Headstage** option only if you are using a ZC-SW switching headstage. If the electrode has a reference site, you may want to short reference to ground in the headstage to help with stimulation artifacts. See Using a Switching Headstage for more information.

> **Caution**
>
> The switching headstage only supports up to ±15 V per channel and can NOT be used in Serial Output Mode.

Option	Description
Output Type	Select whether this sub amp is current-controlled or voltage-controlled. AC Mode adds a DC block to the output signal (an in-line 20 MOhm resistor and 22 uF capacitor in parallel).
Output Mode	The settings in this group determine how the input VoiceIn signals are mapped to a particular hardware voice on a board of the sub-stimulator. It handles this channel mapping for you, so the stim channels you give it from the Electrical Stim Driver gizmo make sense for how the headstage is physically connected. **Parallel** assumes boards in this sub-stimulator are wired in parallel, so you can get more than 4 unique voices per 16 channels or up to 4 higher current voices per 16 channels. The VoiceIn input to the sub-stimulator must contain enough voices/channels. **Serial** assumes channel 16 on all sub-stimulator boards is physically shorted together. Allows one channel of bipolar stimulation with a higher compliance range. **Repeated** repeats the same input voices on all boards of the sub-stimulator.
Number of Boards	Number of boards in sub-stimulator
Current Doubling	**None** - all four hardware voices on each board are unique. **By two** - connect pairs of hardware voices together on each board to achieve 2x max current compliance. **By four** - connect all four hardware voices together on each board to achieve 4x max current compliance.
Voices per Board	How many unique stim signals for each stimulator board
Range per Board	How many channels are available to stim on each board

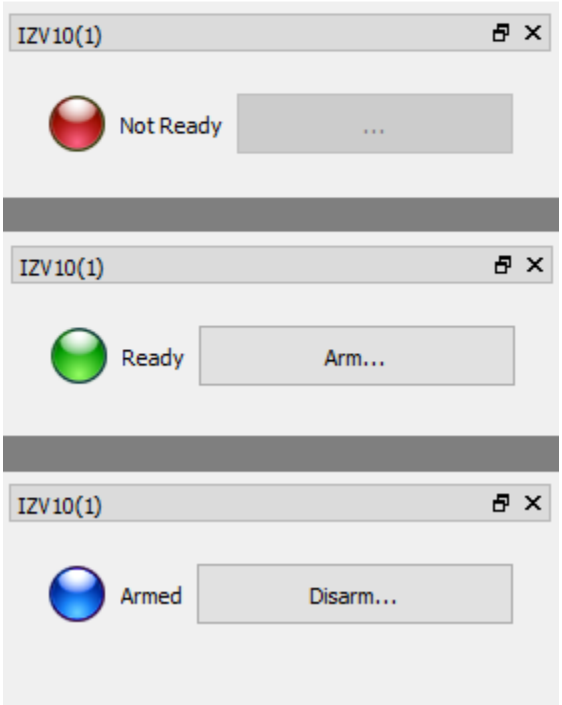


IZV Stimulator Compliance and Safety Tab Options

Option	Description
Safety Mode	If enabled, IZV checks total output current and will disarm if system is outside specs. The device requires arming before it will operate. To arm, run your experiment and make sure no stim presentation is sent to the device. The physical IZV card will light all LEDs red if it is not ready to be armed. It will like all LEDs green when it is ready to be armed. When armed, it will light the LEDs on the stimulation channels during stimulation.
Remote Arming	Enable this to force the user to arm the device via the SIM touchscreen interface. Otherwise, the user can arm the device via a button in Synapse at run-time.
Max Output Voltage	Set the maximum allowed output voltage for this sub-stimulator. Max is 15-18 V, depending on the current battery level. If Safety Mode is Enabled and any channel goes beyond 1 V of this maximum, a Fault occurs and disarms the device.
Grounding	Isolated - each bank of 16 uses its own ground connection for return path. Shared - all banks in sub-stimulator share the same ground return path (ground on first bank of sub-stimulator). To Banana Jack - connect sub-stimulator ground to the external banana jack on physical SIM device. **Caution:** When using multiple sub-stimulators make sure they aren't all sharing the External Ground connection or else they won't be isolated!
Monitoring	Off - disable compliance voltage monitoring. View Only - visualize the compliance voltage for each voice in the Flow Plot at runtime but do not save it to disk. Store - visualize and save the compliance voltage for each voice. The voltage data is scaled by 1000 and saved as 16-bit integers at the system sampling rate.
Impedance Check	Enable run-time impedance check
Save to CSV	Log impedance values into CSV file stored within the block folder whenever an impedance check runs

See the SIM section of the System 3 Manual for more information about configuring and using the IZV interface.

Runtime Interface



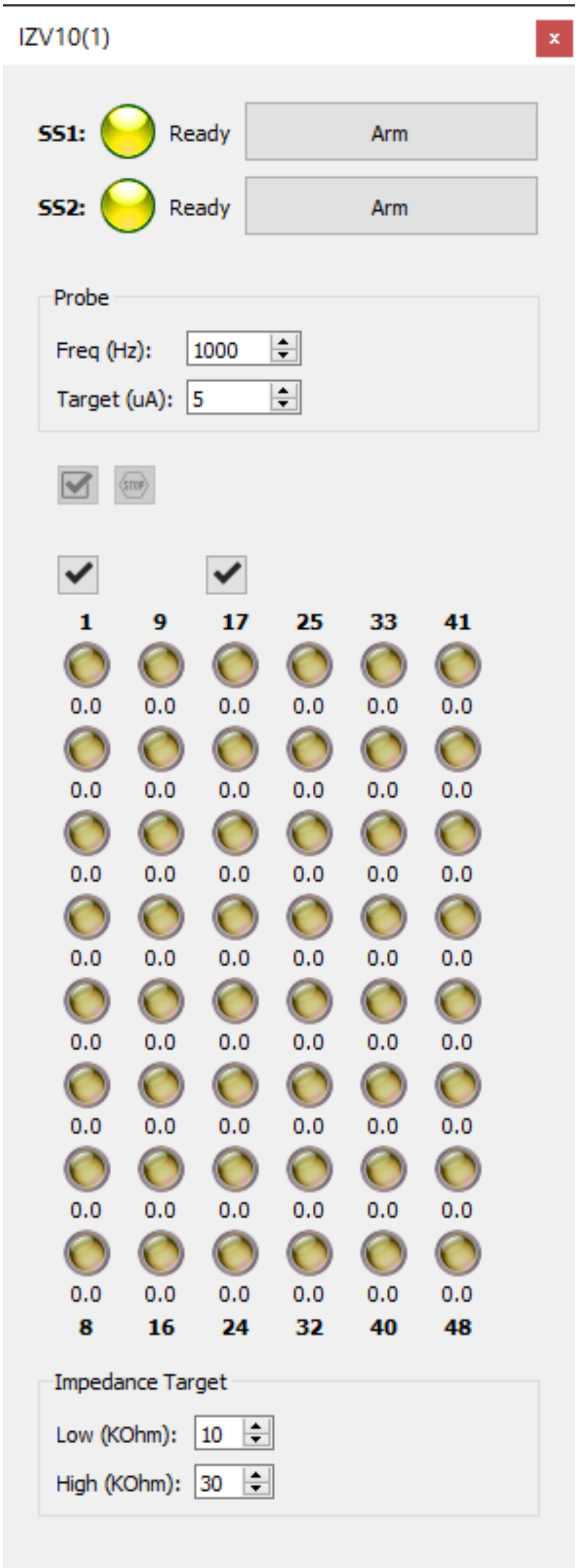
States of the IZV Runtime Interface

If **Safety Mode** is enabled on the Safety and Compliance Tab, then an LED will show the state of the compliance monitor. If Remote Arming is disabled, the Arm/Disarm button is shown and the device can be armed when ready.

The system can only Arm if the fiber is connected and a zero current command is sent to the SIM device, so make sure the gizmo controlling stimulation (typically an Electrical Stim Driver) is muted.

Impedance Checking

When **Impedance Checking** is enabled, a user interface appears at runtime.



IZV Impedance Checking Interface

The IZV tab provides an interface for impedance checking on all channels. The display represents the stimulation channels divided into banks of eight channels, broken up by sub-stimulator. Channel numbers are labeled above and below the bank column in the diagram.

In the area above the probe diagram, you can set test signal frequency and amplitude, and define the high and low impedance threshold targets for visualization below the diagram.

Running the Check

Run an impedance check on the selected sub-stimulator. The test signal (sine wave of frequency defined by Freq (Hz) parameter and amplitude defined by Target (uA) parameter) is presented iteratively on each channel in the currently selected sub-stimulator for 500 ms and the impedance is measured.

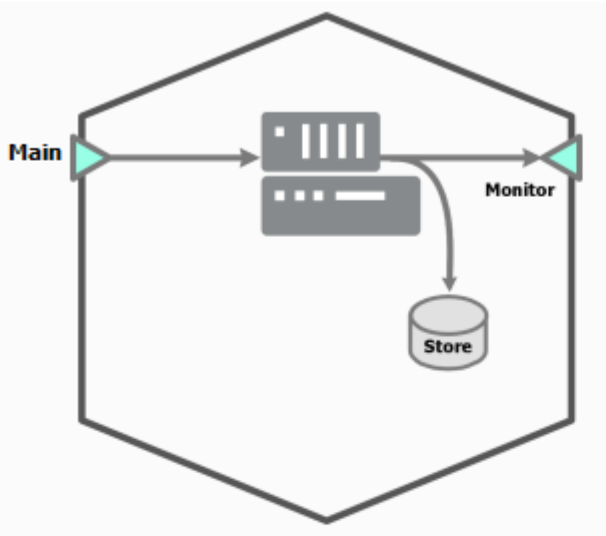
Check All. Run an impedance check for all channels on all sub-stimulators by cycling through each bank of channels using the test signal as described above.

Stop Checking. Stops the impedance checking prematurely.

Results of impedance check are indicated by color: below low impedance threshold (green), above high impedance threshold (red), between low and high impedance thresholds (yellow). The actual impedance values (in kOhm) are displayed beneath each indicator.

Double-click on an individual LED indicator to run the impedance check on just that channel.

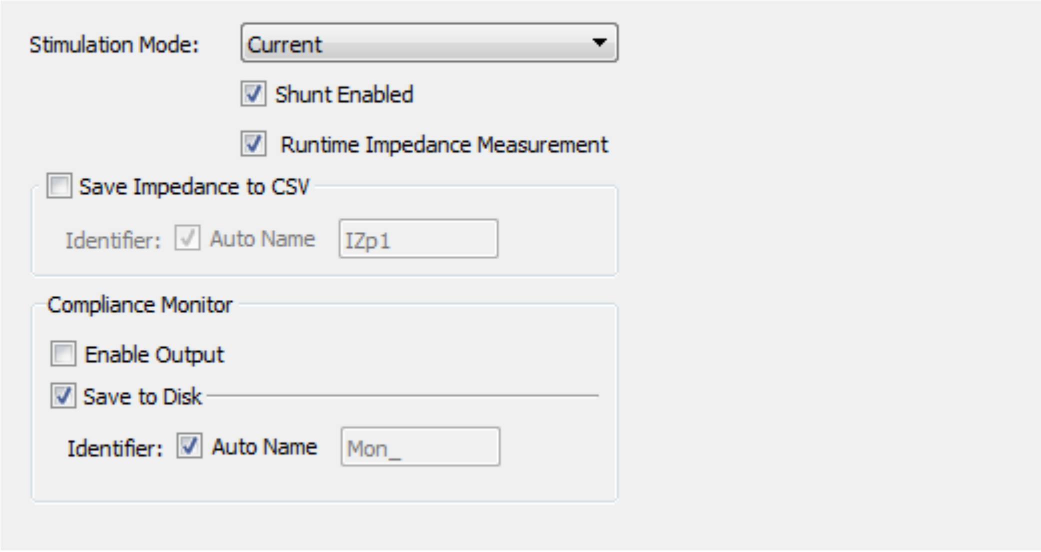
IZ2 Electrical Stimulation



IZ2 Block Diagram

The IZ2 requires a multi-channel floating point data source.

IZ2 Options



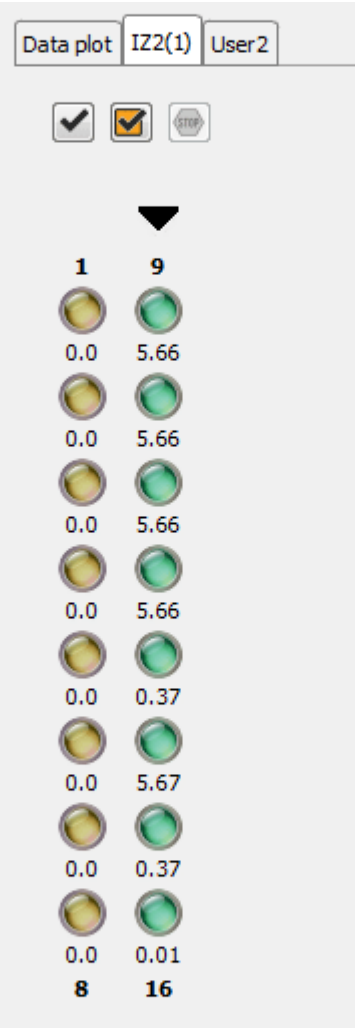
IZ2 Options

Option	Description
Stimulation Mode	Select current or voltage (IZ2/IZ2H only) mode from drop-down menu
Save Impedance to CSV	Log impedance values into CSV file stored within the block folder, whenever an impedance check runs either manually or through the SynapseAPI.
Compliance Monitor	This represents the actual voltage on the output of the stimulator for the currently selected bank (see Runtime Interface, below). Check **Enable Output** to make this a data source other gizmos can use.

See the Stimulus Isolator section of the System 3 Manual for more information about stimulator operations.

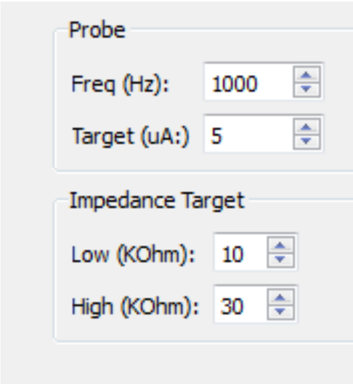
Runtime Interface

When **Runtime Impedance Measurement** is checked, a user interface appears at runtime.



The IZ2 tab provides an interface for impedance checking on all channels. The display represents the stimulation channels divided into banks of eight channels. Channel numbers are labeled above and below the bank column in the diagram. The currently selected bank is indicated by a **black arrow** above it. This is also the bank that is actively monitored in the Compliance Monitor. To change the selected bank, click any LED in the desired bank.

Defining the Probe and Target



In the area beneath the probe diagram, you can set test signal frequency and amplitude, and define the high and low impedance threshold targets for visualization.

Running the Check

Run an impedance check on the currently selected bank. The test signal (sine wave of frequency defined by Freq (Hz) parameter and amplitude defined by Target (uA) parameter) is presented iteratively on each channel in the currently selected bank for 500 ms and the impedance is measured.

Check All. Run an impedance check for all channels by cycling through each bank of eight channels using the test signal as described above.

Stop Checking. Stops the impedance checking prematurely.

Results of impedance check are indicated by color: below low impedance threshold (green), above high impedance threshold (red), between low and high impedance thresholds (yellow). The actual impedance values (in KOhm) are displayed beneath each indicator.

iCon Behavioral Control Interface

Overview

The iCon is a behavioral control interface designed to support easy or complex behavioral experiments. It has interchangeable analog and digital modules to fit the needs of any experimental paradigm. It can control most third-party behavioral chambers. The iCon connects directly to any TDT RZ processor with a DSP-M or optical quad DSP card and is controlled with TDT's Synapse software — resulting in high-bandwidth, precision control of behavioral hardware with real-time signal monitoring and full integration with neural recordings.

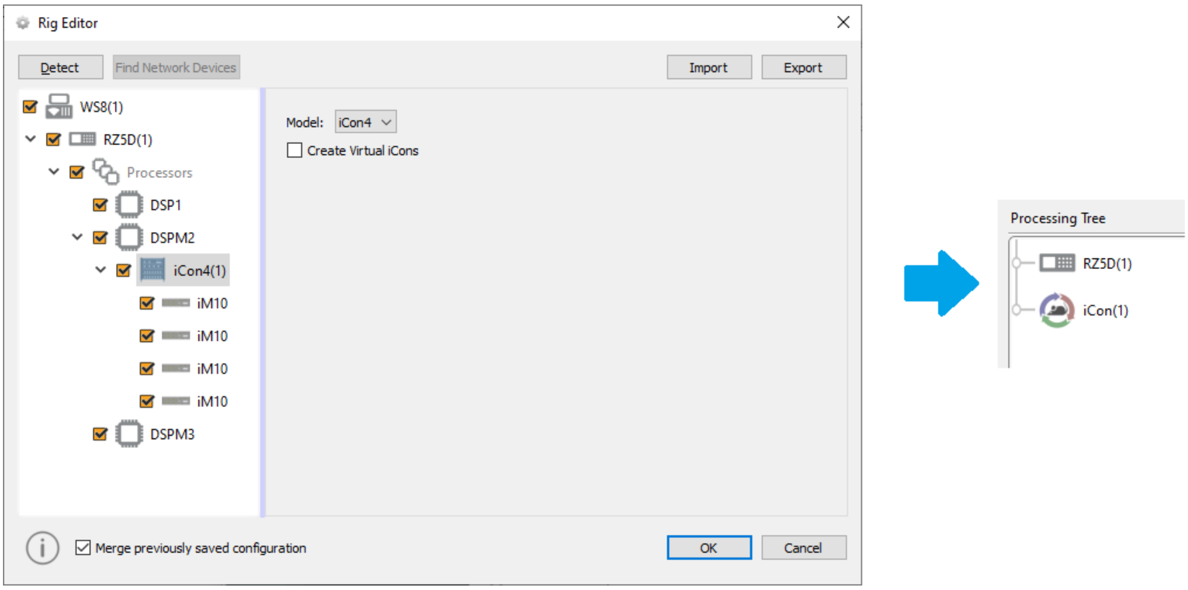
The iCon is available with two (iCon2) or four (iCon4) module slots. It is configurable with the following modules:

- iA4 Differential Bioamp
- iD2 Digital Headstage Interface
- iHn High Voltage Interface
- iL24 Digital Logic Interface
- iMn Multi-function Interface
- iRn IR Driver Interface
- iS9 Aversive Stimulator
- iVn Video Capture Interface
- iXn Lux Optical Interface

See the Hardware Manual for information on hardware setup and technical specifications.

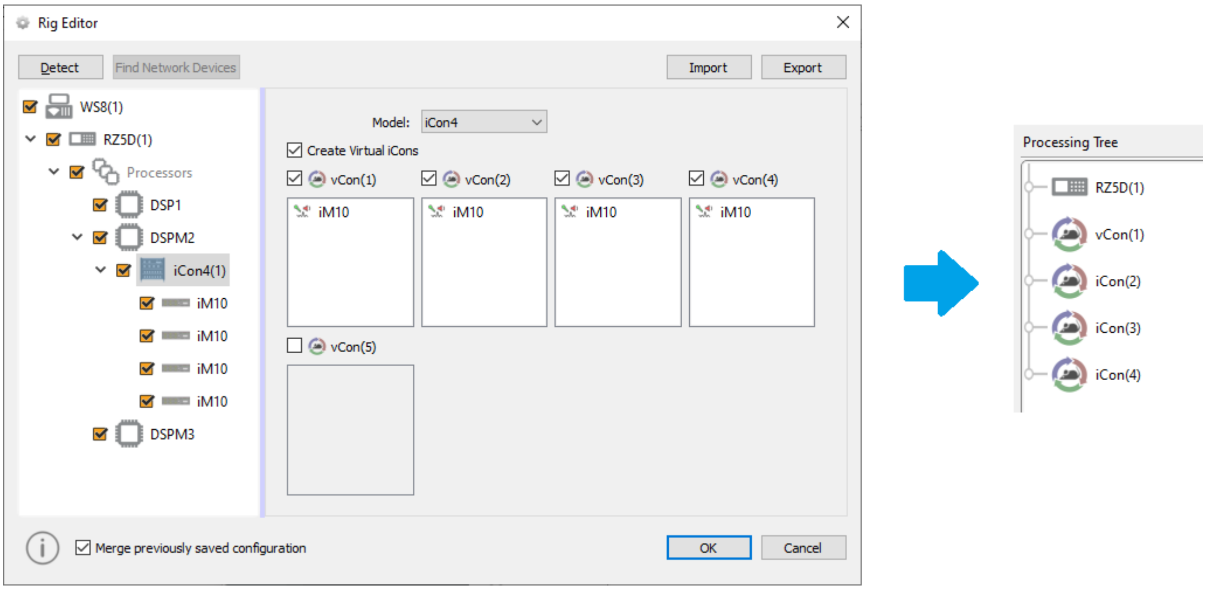
Rig Setup

When attached to a DSPM card, the iCon is automatically detected and populated in the Synapse Rig Editor. By default, each iCon is its own object in the Synapse Processing Tree.



Icon detected, Processing Tree with iCon

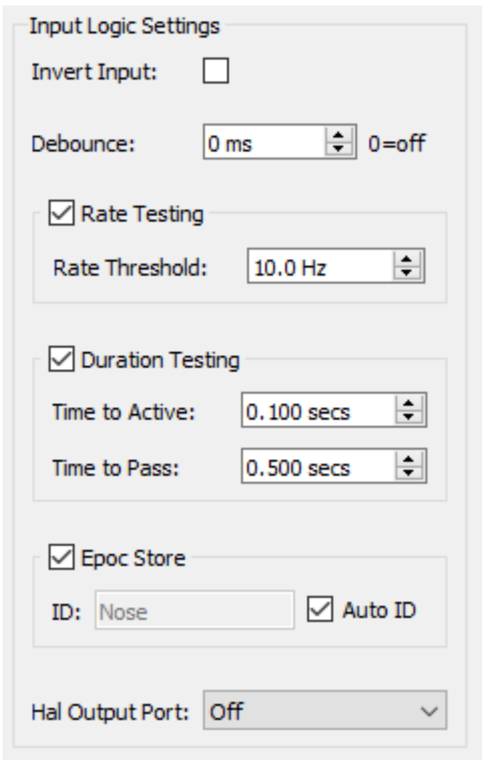
By default, all iCon submodules are attached to a single iCon object in the Processing Tree. You may wish to logically split modules from their parent iCon and control them independently. For example, if you are running four separate cages with a single iCon4 that has four iH8 modules, you would want them available as individual objects in the Processing Tree for independent control.



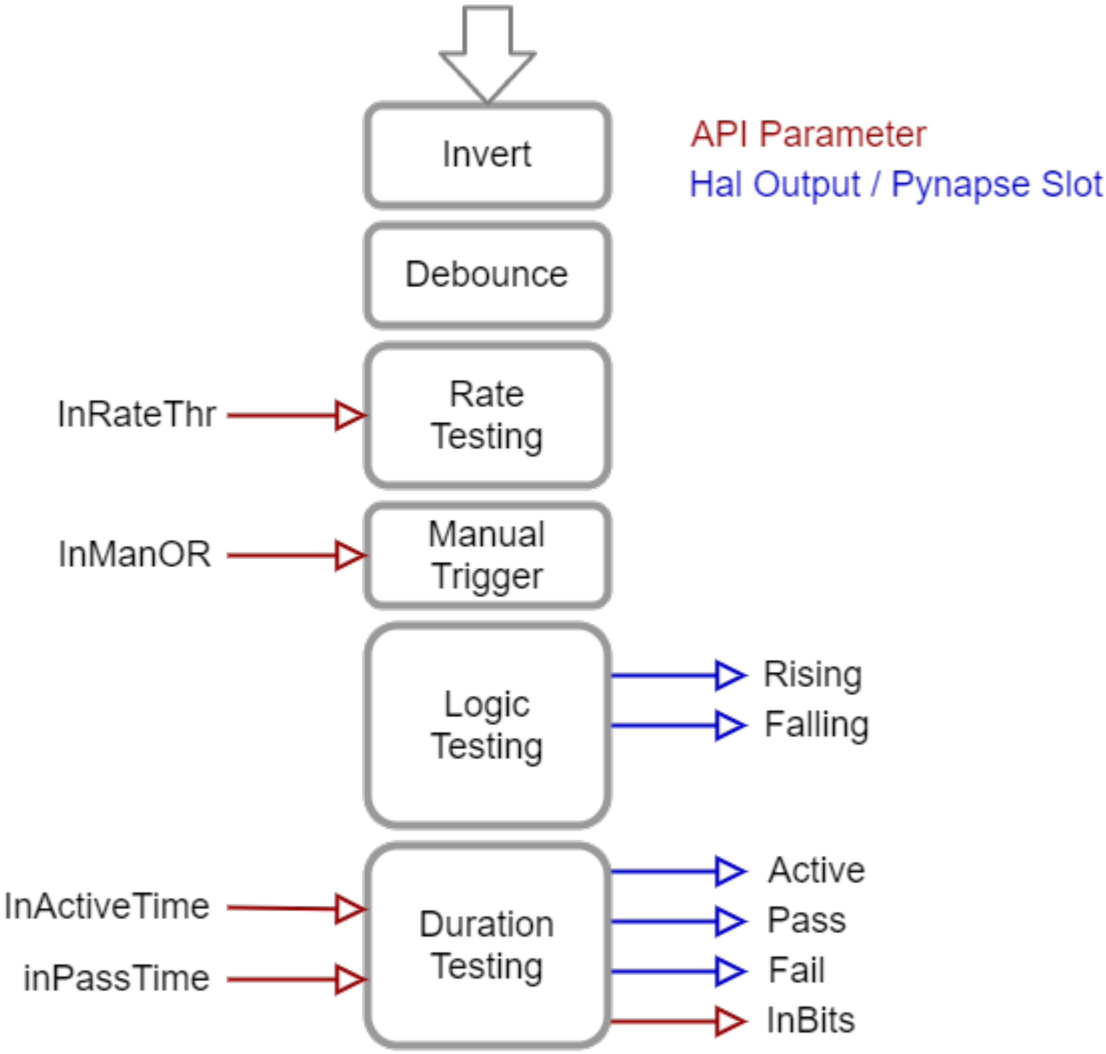
Icon configured as four virtual iCons, Processing Tree with four virtual iCons

Logic Input Processor

The iHn inputs, iL24 inputs, iRn inputs, and iMn and iS9 touch sensor inputs are all digital signals (logic TTL) that go through an input processor. The iMn analog inputs first go through an additional analog input processor, see iMn Input Processor below for more details.



Logic Input Options



Logic Input Processor

Debounce is the amount of time the input has to settle before its new state is used. This is useful for lever presses or hardware button presses which can 'bounce' on contact and trigger several rapid artificial events before making solid contact.

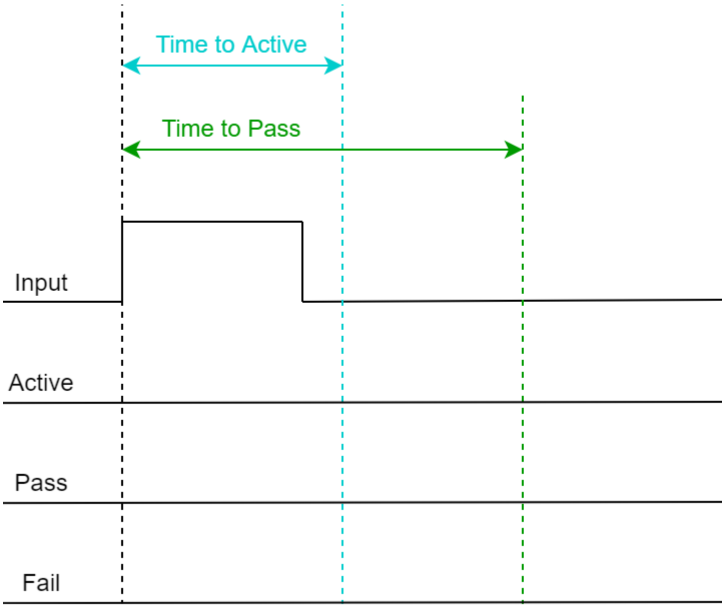
You can optionally save epoch timestamp events for each input. An integer code for the event type is stored with the timestamp. See Epoc Storage below for more information.

Duration Testing

The inputs can use built-in duration testing. In this example, the button has to be pressed for 600 ms to get to the 'Active' state, and another 400 ms for it to 'Pass'.

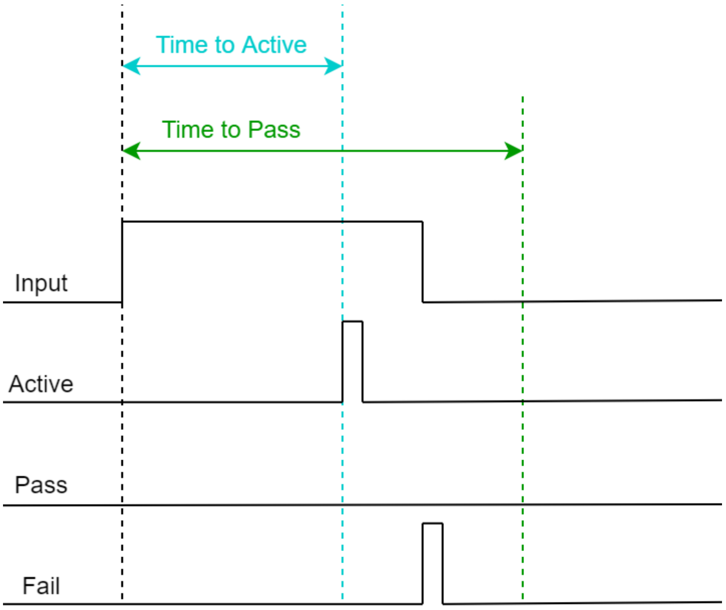
No Trial

Time to Active not reached by Input.
Pass and Fail not activated.



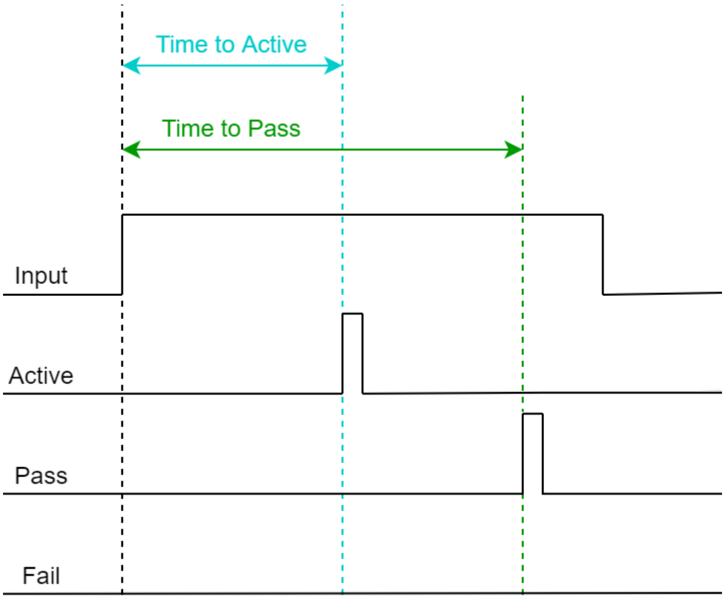
Fail

Time to Active reached by Input, so 'Active' trigger fires.
Time to Pass was not reached by Input, so the 'Fail' trigger fires when Input is released.



Pass

Time to Active reached by Input, so 'Active' trigger fires.
Time to Pass also reached by Input, so 'Pass' trigger fires.



Epoc Storage

Epoc events are triggered on the 'rise' event of the input and a timestamp and value of 3 is stored in the data tank. If the input is true for more than 2 samples then the 'fall' event is also timestamped and stored, with a value of 4.

The full state of the input, including duration test results, is captured in the integer code:

b5	b4	b3	b2	b1	b0
Fail	Pass	Active	Fall	Rise	True

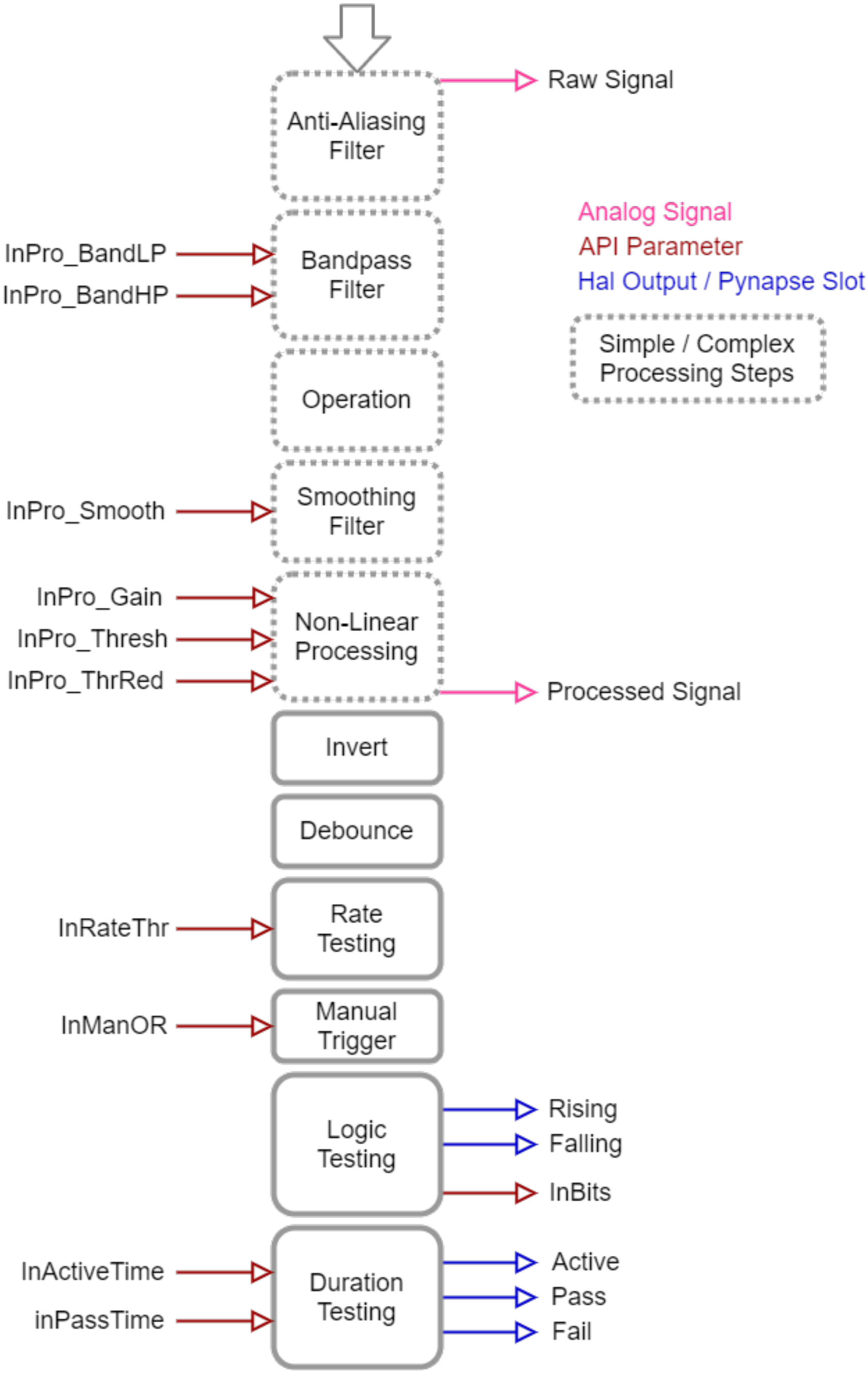
Example values of the epoc event:

Event	Value	Binary Representation
Rise	3	0x0000011
Fall	4	0x0000100
Active	9	0x0001001
Pass	17	0x0010001
Fail	36	0x0100100

Connect the result of the input processor to other gizmos by enabling the **Hal Output Port**. You can connect any of these bits, or the integer (word) value, to the output port.

iMn Input Processor

The iMn analog inputs uses an additional input processor before the Logic Input Processor. The entire input processor for the iMn is shown below.

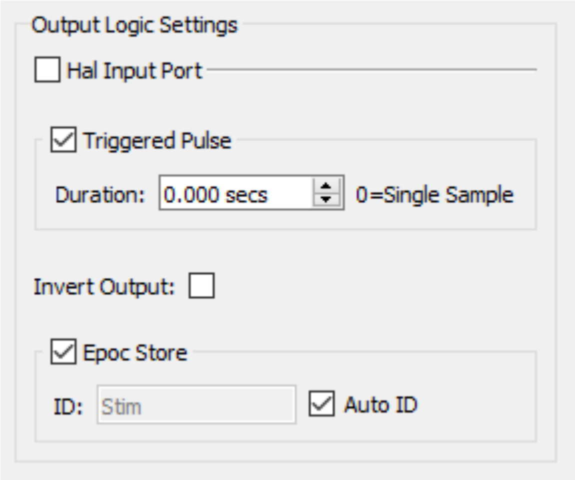


iMn Input Processor

The raw and processed analog signal is available as a Hal output link to optionally view and save the signal. This is useful for debugging the analog processing.

See the iMn Synapse Manual for more information.

Logic Output Processor



Logic Output Options

All outputs are digital signals (logic TTL) that are either turned on/off, triggered for a single sample, or strobed high for a fixed duration.

Set the **Name** of the output to something that makes sense for your experiment, e.g. 'Reward'.

You can optionally save epoch timestamp events for each output. A timestamp is saved when the output turns on. If the output is high for more than 2 samples then the offset is stored as well.

Triggered Pulse - output stays high for a fixed amount of time. If **Duration** is 0, this is a single sample. If **Duration** is greater than 0, use the `OutDur` API parameter to dynamically set the duration.

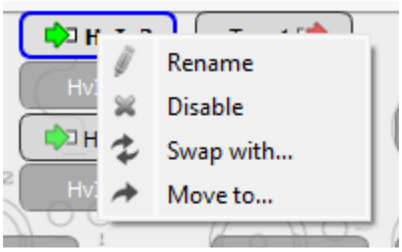
The output can be controlled by a **Hal Input Port** link, setting the `OutEnab` API parameter, or by clicking the output on the Runtime Monitor.

iMn Output Processor

The iMn uses the signal from the logic output processor to generate an analog waveform. See the iMn Synapse Manual for more information.

Design Interface

Each iCon module's I/O is assigned through buttons on the user interface that match the physical location on the module. If you create an I/O that you want to move to a different physical location on the iCon, right-click and use the "Swap with..." or "Move to..." options. The possible targets for the swap or move will be highlighted. You can also select a target on a different module within the same iCon.



WordIn / WordOut

All inputs states (true, pass, fail, etc) are packed into an integer that is available with the `InBits` API parameter for each input. The WordIn settings let you pack the raw input states (true/false only, directly from hardware before any processing) into an integer and make it available as an output link that other gizmos can connect to.

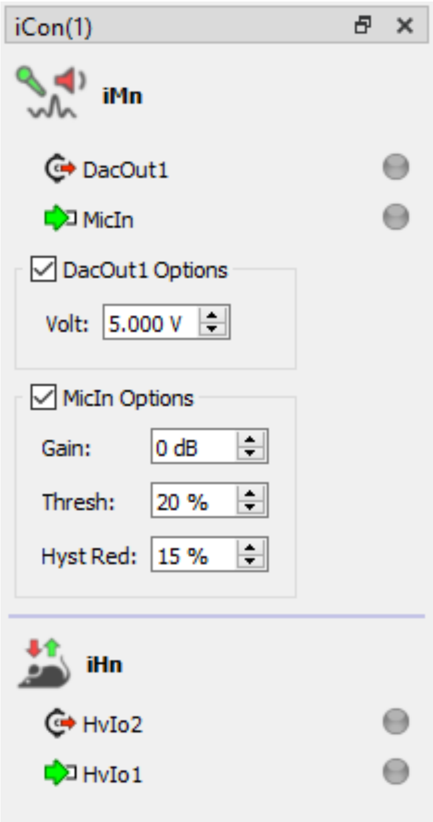
The `OutWord` API parameters lets you set all raw output states with a single integer. The WordOut settings let you optionally control all iHn outputs or iL24 outputs with a single gizmo input.

Pynapse Integration

The iCon can be directly controlled by Pynapse. The same input/output settings are available from within the Pynapse user interface. In this case the WordIn / WordOut are not available.

Runtime Interface

When **Runtime Monitor** is checked, a user interface appears at runtime.



The iCon Tab shows you the state of the inputs/outputs, and lets you manually trigger the inputs/outputs by clicking them.

iAn Bioamp

The iAn Bioamp is a 4, 16, or 32 channel amplifier for neurophysiology recordings. The iAn device models support a wide range of recording modalities for a variety of frequency and electrode impedance ranges.

iA4 Differential Bioamp

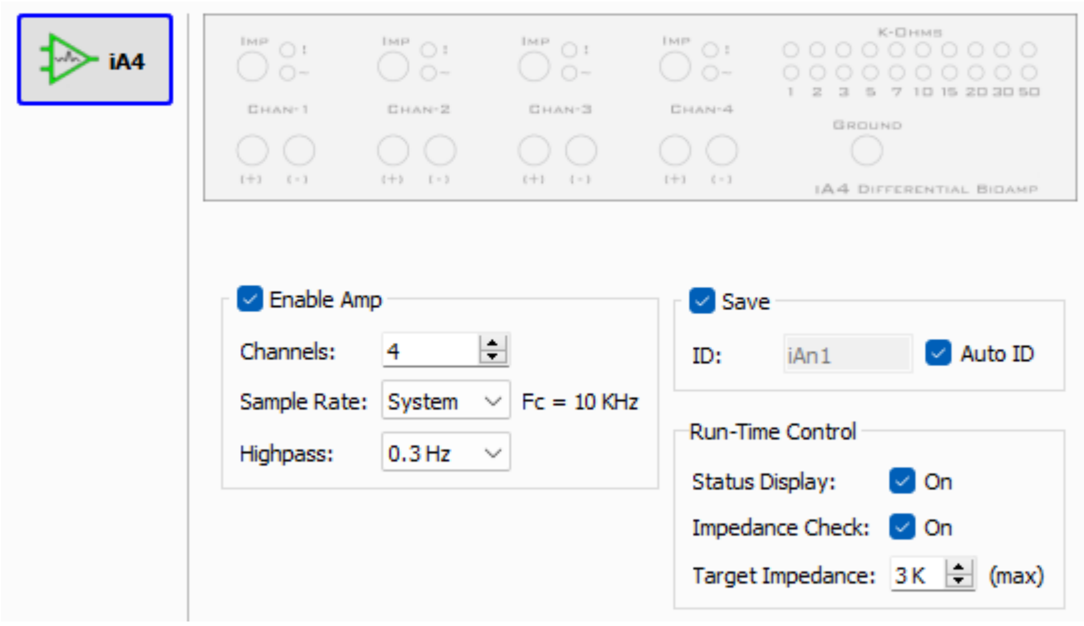
The iA4 is a four channel differential low-noise bioamp targeted for use in low channel count, low impedance neurophysiology recordings such as ABR, EEG, EMG, or EKG signals.

Connections to the subject (up to four pairs of channels and ground) are made with standard 1.5 mm touchproof safety connectors.

- The iA4 includes impedance testing functionality for each of the four recording channels (+) and indifferent channels (-)
- Manual impedance check buttons simplify testing and debugging of connected electrodes
- Onboard LEDs provide live feedback of the recording (+) and indifferent (-) channel impedances

See the Hardware Manual for information on iA4 technical specifications.

iA4 Options



iA4 Interface

Select the number of **Channels** (differential pairs) to read from the iA4.

There are selectable **Sample Rate** options of ~750 Hz, ~1.5 kHz, ~3 kHz, ~6 kHz, ~12 kHz, ~25 kHz, ~50 kHz or system rate (maximum ~50 kHz). The lowest practical rate improves noise performance.

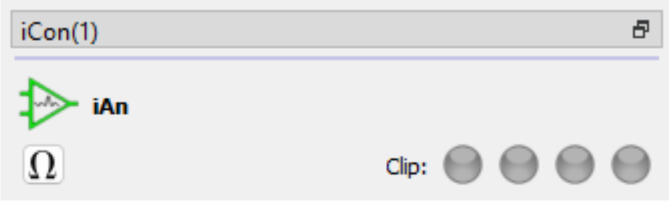
A user-selectable **Highpass** filter has adjustable options that are: 0.3 Hz, 1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, 64 Hz.

The **Save** option visualizes and stores the raw signal at run-time.

The **Status Display** option allows the user to enable the clipping indicators for each channel at run-time.

The **Impedance Check** option enables online impedance checking. Specify the **Target Impedance** threshold for the red/green runtime indicator for each channel.

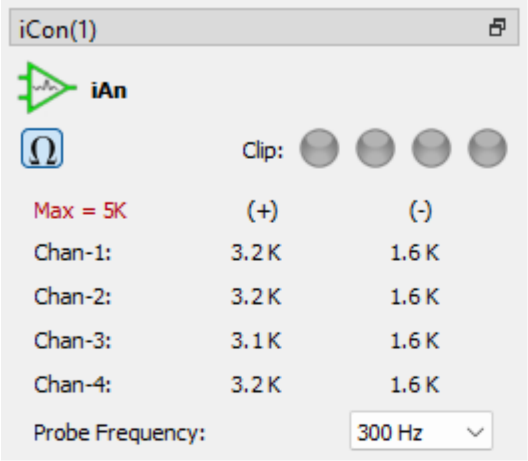
iA4 Runtime Interface



iA4 Runtime Interface

There are **Clip** indicators on each channel to indicate if the signal is within ~1.5 dB of max input range.

Click the **Impedance** button to switch the iA4 into impedance checking mode.



iA4 Impedance Check Interface

Users can select a **Probe Frequency** of 100 Hz, 300 Hz, 500 Hz, or 1 kHz for the impedance check. Take care that you do not choose a frequency that is beyond the Nyquist frequency (~½ the sampling rate defined during designtime).

The (+) and (-) channel impedances are alternately measured every ~110 ms.

iA16 / iA32 Bioamp

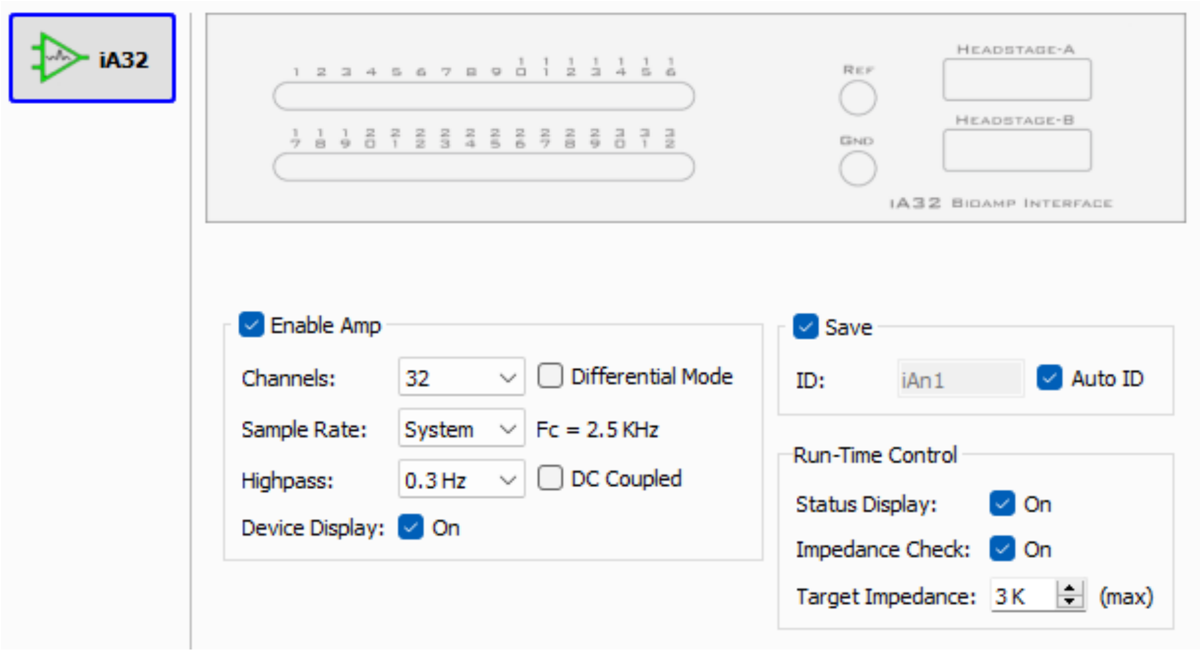
The iA16 and iA32 are 16 and 32 channel low-noise bioamps that can operate in shared or true differential mode. They can record signals all the way down to DC.

DB26 connectors mate with TDT's standard headstages to connect to the subject.

- The iA16/32 includes impedance testing functionality for each of the recording channels and reference
- Onboard LEDs provide live feedback of the recording channel activity and impedances

See the Hardware Manual for information on iA16 / iA32 technical specifications.

iA16 / iA32 Options



iA32 Interface

Select the number of **Channels** (differential pairs) to read from the iA16 / iA32. By default, it uses a shared reference input. Optionally select **Differential** to record true differential where each even channel is the reference for the preceding odd channel.

There are selectable **Sample Rate** options of ~750 Hz, ~1.5 kHz, ~3 kHz, ~6 kHz, ~12 kHz, ~25 kHz, ~50 kHz or system rate (maximum ~50 kHz). The lowest practical rate improves noise performance.

A user-selectable **Highpass** filter has adjustable options that are: 0.3 Hz, 1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, 64 Hz. The iA16 / iA32 can optionally be **DC Coupled** as well.

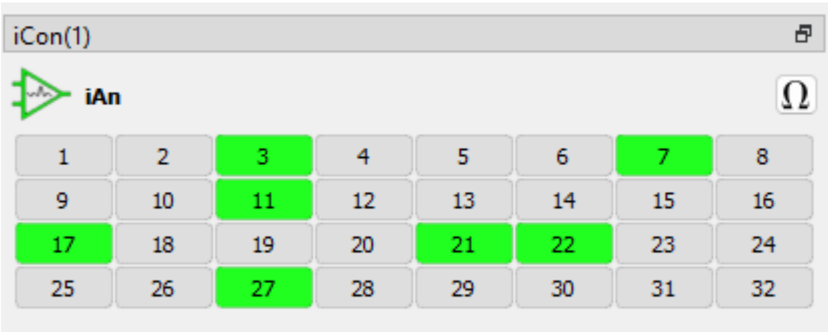
The **Save** option visualizes and stores the raw signal at run-time.

The **Device Display** option enables the clipping / activity indicators on the front panel of the module.

The **Status Display** option allows the user to enable the clipping indicators for each channel at run-time.

The **Impedance Check** option enables online impedance checking. Specify the **Target Impedance** threshold for the red/green runtime indicator for each channel.

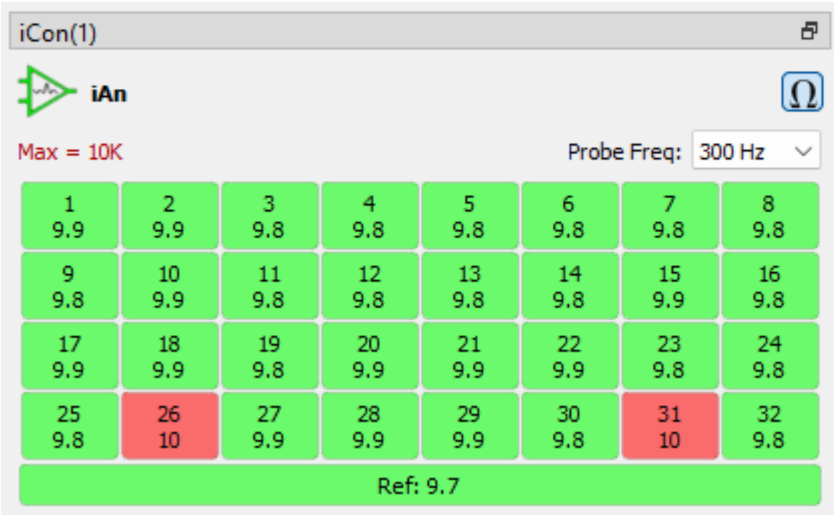
iA16 / iA32 Runtime Interface



iA16 / iA32 Runtime Interface

At 25 kHz sampling rate and above, the channel indicators flash green when there is spike activity. On the front panel of the iA16 / iA32, the channel LEDs are dim green when enabled, and flash brightly with spike activity.

Click the **Impedance** button to switch the iA16 / iA32 into impedance checking mode.



iA32 Impedance Check Interface

Users can select a **Probe Frequency** of 100 Hz, 300 Hz, 500 Hz, or 1 kHz for the impedance check. Take care that you do not choose a frequency that is beyond the Nyquist frequency (~½ the sampling rate defined during designtime).

The channels and reference are alternately measured every ~250 ms.

iD2 Digital Headstage Interface

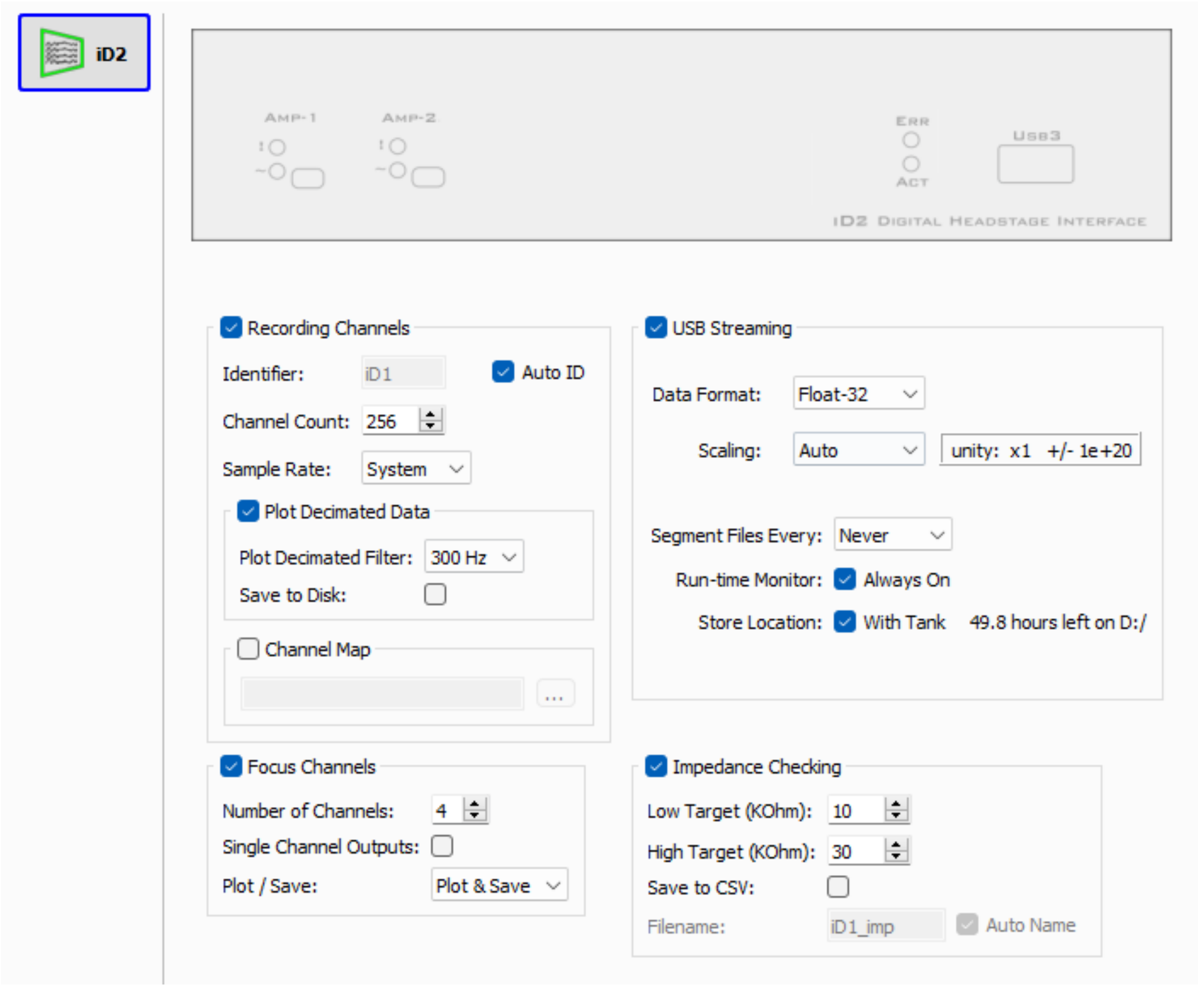
The iD2 is an interface to Intan-based digital headstages. It has two input connectors that can read up to 128 channels each, for a total of 256 channels. All raw data is streamed via USB 3 back to the computer running Synapse and recorded straight to disk. A compressed signal of all 256 channels is returned to Synapse for visualization. Up to 16 selectable channels of the raw data are also available on the DSP for real-time processing and visualization.

- The iD2 includes impedance testing functionality for each headstage
- Onboard LEDs provide live feedback of the headstage connection and activity

See the Hardware Manual for information on iD2 technical specifications.

> **Important**
>
> The iD2 USB Streaming capability is not compatible with the iPac and SynCon software.

iD2 Options



iD2 Interface

Recording Channels

Select the total **Channel Count** to return from the iD2. This is the sum of all headstages that are connected to the iD2 that you want to read.

There are selectable **Sample Rate** options of ~750 Hz, ~1.5 kHz, ~3 kHz, ~6 kHz, ~12 kHz, ~25 kHz, or system rate (maximum ~25 kHz). The lowest practical rate improves noise performance.

Enable **Plot Decimated Data** to return a highly compressed version of the raw signal to the processor and Synapse for visualization of all headstage channels. The plot decimated data consists of the max and min values of a short chunks of points. The rest of the chunk is thrown away. The data rate depends on the System rate and number of channels, but is usually ~300-1500 Hz. This is fine for monitoring the signal for activity and requires very low bandwidth, but is not useful for data analysis.

The plot decimated data can optionally be filtered on the iD2 itself. If the iD2 is running at ~6 kHz or above, you can enable a 100 Hz, 300 Hz, or 500 Hz filter on the device. At lower sampling rates, there is no filter available.

At runtime, you can use the **Plot Decimated Filter** cutoff frequency as a highpass, as a lowpass, or disable it. See Runtime Interface below.

Set a **Channel Map** to rearrange the channels on the iD2 before they are returned to the processor. The format of the CSV file for specifying the channel map is:

```
{Synapse Channel},{Headstage Channel}
{Synapse Channel},{Headstage Channel}
```

You do not have to map all the channels, and you can list the map in any order/

Focus Channels

Up to 16 user-selectable **Focus Channels** are returned to the processor and Synapse at the full data rate for real-time processing and visualization.

Optionally create **Single Channel Outputs** to link to other gizmos, depending on the number of focus channels you have.

Use the **Plot / Save** option to visualize and/or store the raw signal from the focus channels at runtime.

USB Streaming

The front panel of the iD2 has a separate USB3 connection back to the computer that is running Synapse. This connection is stream all channels at full bandwidth straight to disk without going through the processor interface bottleneck.

By default, the iD2 uses the Sample Rate defined in the Recording Channels section. You can select the data format and any scaling. If the Sample Rate is set to System, then the Integer-16 format (with Scaling set to 1e6) is preferred. If Sample Rate is lower than the RZ processor clock, you can get higher fidelity with the Float-32 data format.

Segment Files Every lets you split up the data files automatically at the specified interval. The new file will begin exactly one sample after the previous file. This is useful for long recordings where the user wants access to the raw data for parallel processing before the recording is over.

Run-time Monitor adds a runtime interface that warns you of data errors and USB connection issues. It shows the internal buffer status, data saved, and space remaining.

By default the data is stored in the same block folder as the rest of the recording. Use the **Store Location** option to save the iD2 data files to a different folder.

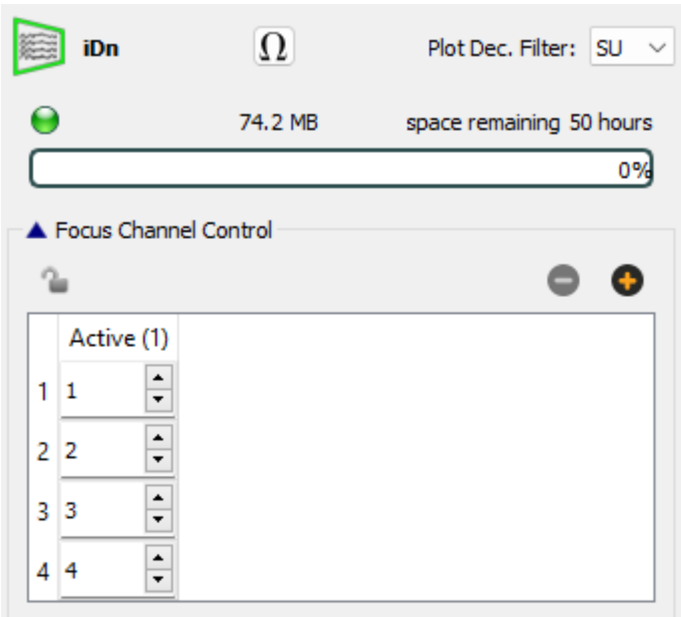
Impedance Checking

The **Impedance Check** option adds a run-time interface for enabling the impedance checking built into the Intan headstage.

Define the **Low Target** and **High Target** impedance thresholds for controlling the color of each channel LED in the runtime interface. This is used to intelligently control the impedance checking circuit on the headstage based on your target impedance.

Save to CSV logs impedance values into a CSV file stored within the block folder, whenever an impedance check runs.

Runtime Interface



iD2 Runtime Interface

Set the state of the plot decimated filter to lowpass (LFP), highpass (SU), or disabled (Off).

USB Streaming Status

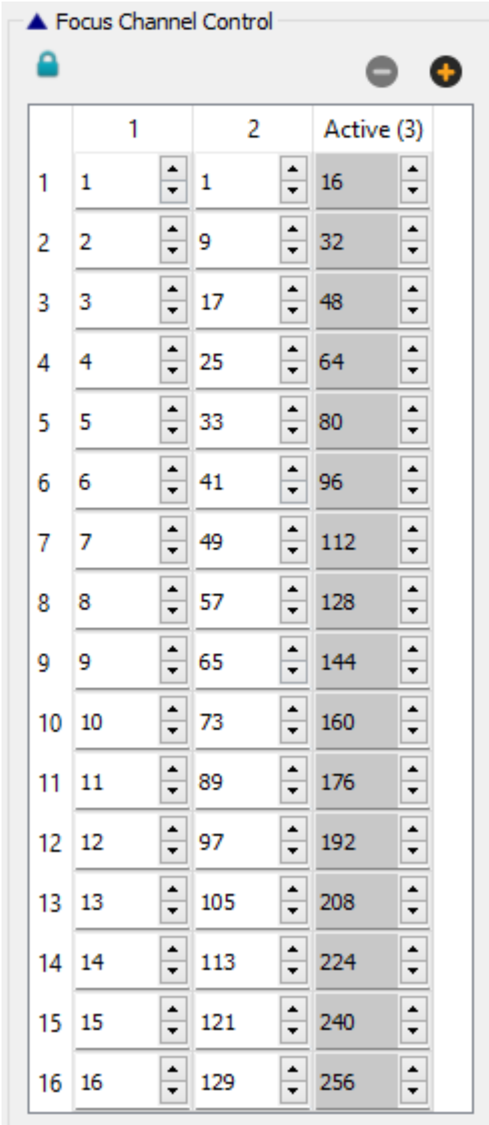
The runtime interface shows the current status of the USB connection. Any errors will turn the LED red and display a message in this box.

The progress bar indicates how much of the iD2 buffer is full of data that hasn't been transferred yet. This will typically be at 0% during runtime, but may spike at the beginning of the recording when the storage files are being generated, or if there are any data transfer issues to the PC.

The runtime interface also shows how much data has been recorded from the iD2, and an estimate of the recording time available on the hard drive.

> **Note**
>
> See the iD2 section of the System 3 Manual for more information about Data Streamer operations.

Focus Channel Control



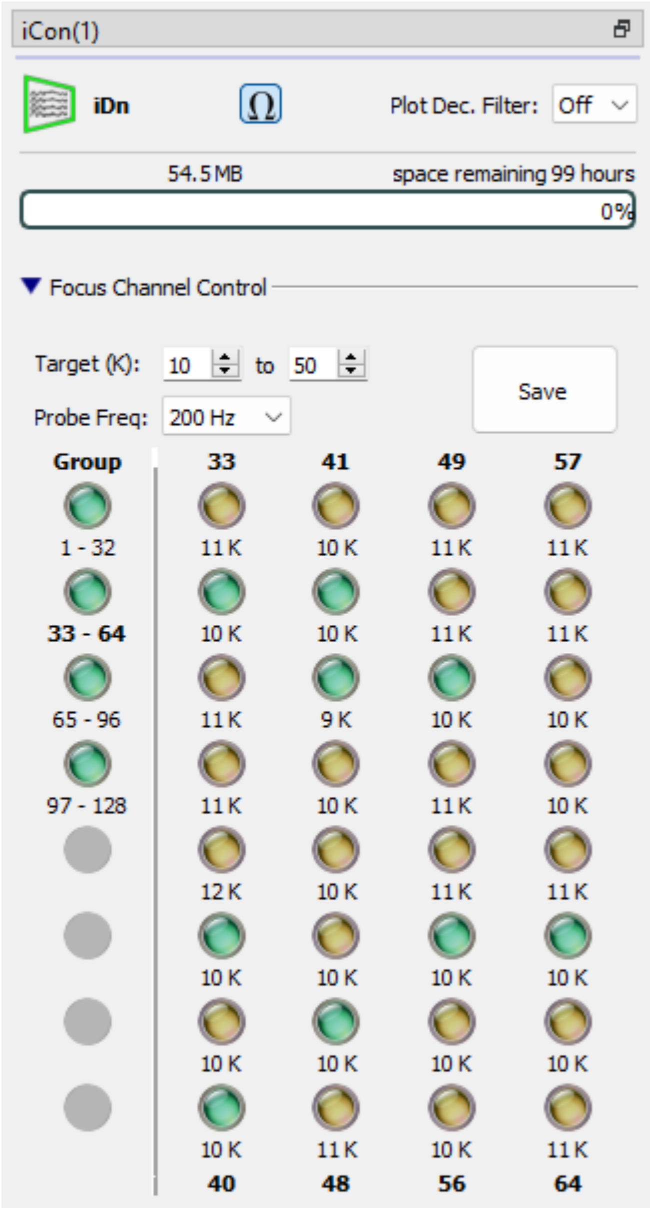
	1	2	Active (3)
1	1	1	16
2	2	9	32
3	3	17	48
4	4	25	64
5	5	33	80
6	6	41	96
7	7	49	112
8	8	57	128
9	9	65	144
10	10	73	160
11	11	89	176
12	12	97	192
13	13	105	208
14	14	113	224
15	15	121	240
16	16	129	256

iD2 Focus Channel Interface

Use the table to create a custom column of channel numbers to include in the focus channel set. You can create multiple columns and double click the column header to choose the "Active" set of channels to return to the focus channel stream. Click the lock icon to lock the currently active column.

Impedance Checking

Click the **Impedance** button to switch the iD2 into impedance checking mode.



iD2 Impedance Check Interface

Users can select a **Probe Frequency** of 24 Hz, 48 Hz, 95 Hz, 191 Hz, 381 Hz, 763 Hz, 1526 Hz, or 3052 Hz for the impedance check. Take care that you do not choose a frequency that is beyond the Nyquist frequency (~½ the sampling rate defined during designtime).

The iD2 loops through the banks of channels and updates the impedance values for each channel, coloring the LED depending on its impedance relative to the high and low target impedances. The Group LEDs display the highest value for each set of 32 channels. Double-click on a Group LED to actively display that set of 32 channels.

Click the **Save** button to log the current impedance values to the CSV file.

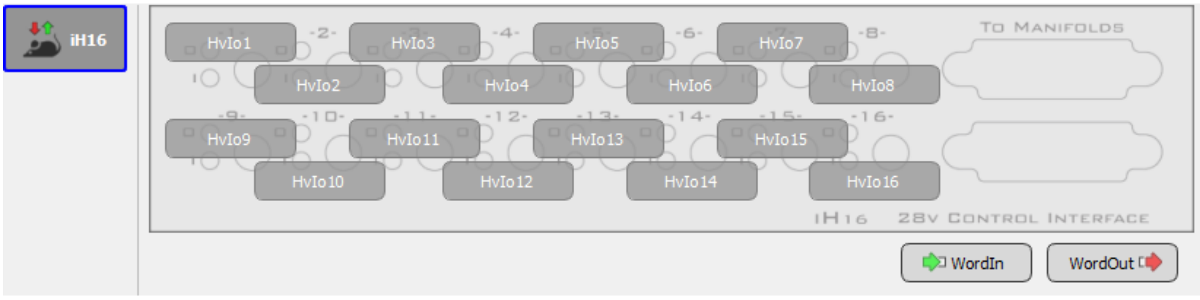
iHn High Voltage Interface

The iH8, iH10, and iH16 feature bidirectional interface ports to provide direct control of high-voltage (e.g. 28 V signal) cage elements including Med Associates and Lafayette.

- The ports are configurable in Synapse as either inputs or outputs
- Manual trigger buttons simplify testing and debugging of connected behavioral components
- Status lights make it easy to monitor component activity during sessions

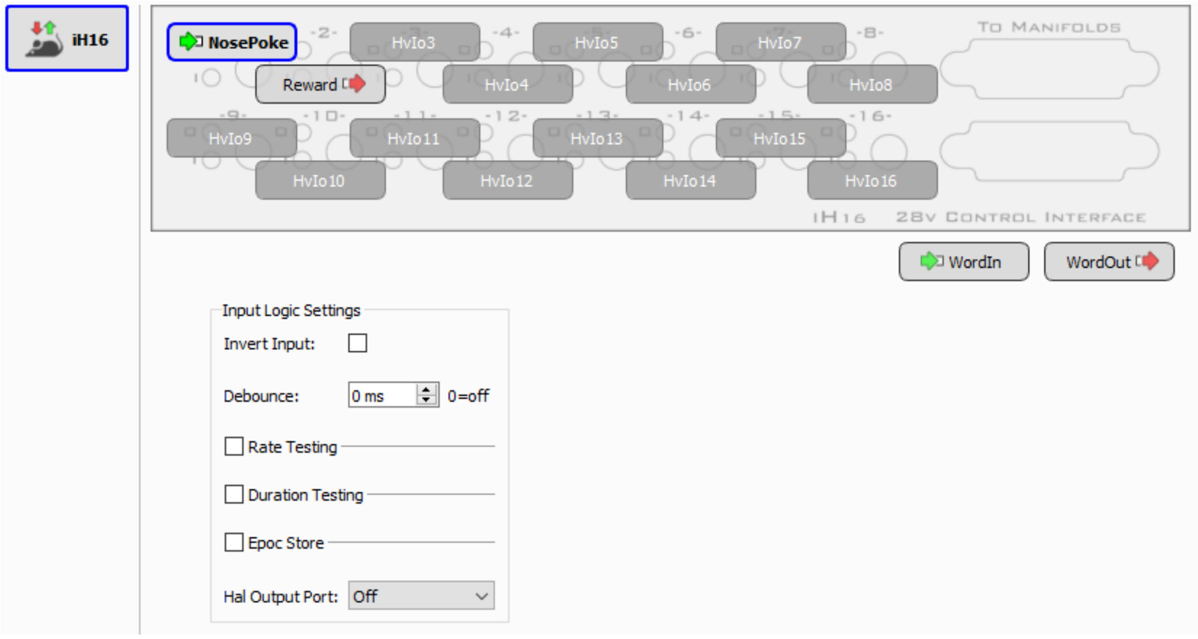
See the Hardware Manual for information on iHn technical specifications.

iHn Options



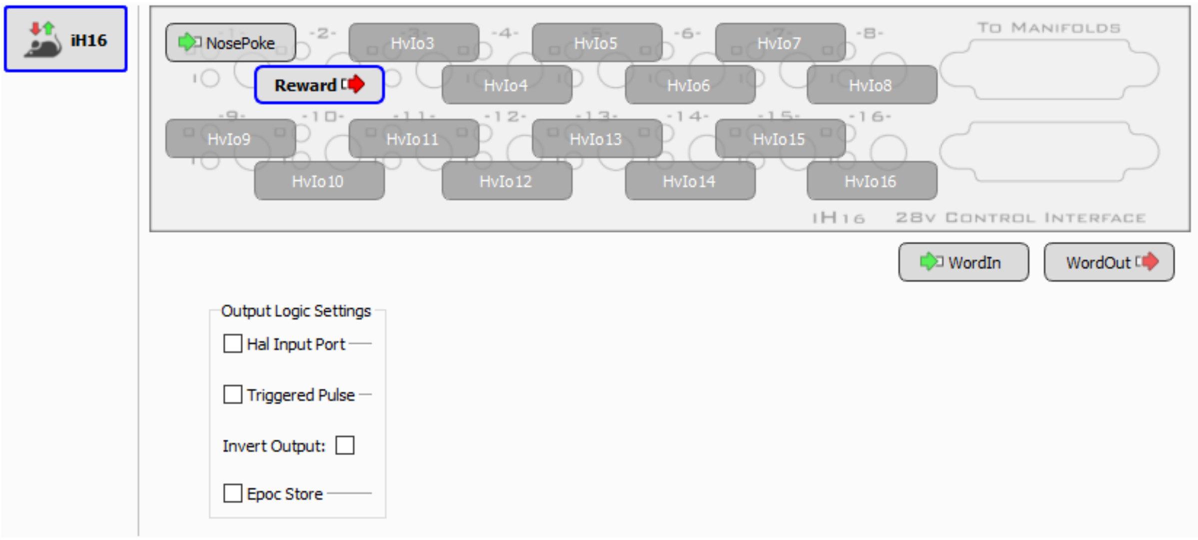
iHn Interface

Click on a port to enable it as an input or output. Set the **Name** of the port to something that makes sense for your experiment, e.g. 'NosePoke'.



iHn Input Options

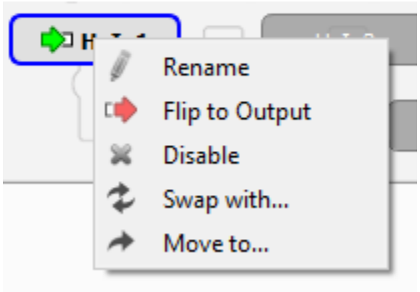
All iHn inputs go through a logical input processor. See Logic Input Processor for more details.



iHn Output Options

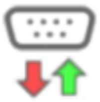
All iHn outputs go through a logical output processor. See Logic Output Processor for more details.

After you define a port, you can modify it or move it by right-clicking on the port.



iHn Context Menu

iL24 Digital Logic Interface

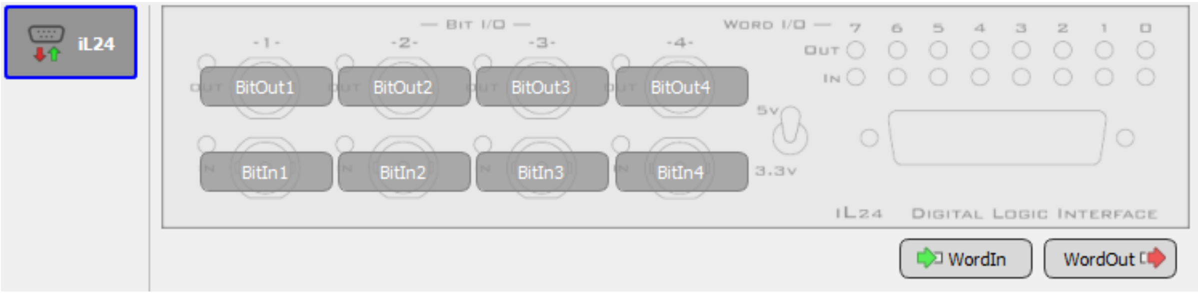


The iL24 module can communicate with external behavioral components using 24 bits of 5 V or 3.3 V TTL logic signals.

- There are four bit-wise inputs and four bit-wise outputs available with BNC connectors and status light for each
- The DB25 connector has access to all 24 addressable bits
- Two rows of 8 status LEDs on the front panel show the state of the word input and output bits
- The front panel switch toggles between +3.3 V and +5 V logic for all 24 bits of I/O on the iL24
- Each bit can source up to 6 mA maximum current

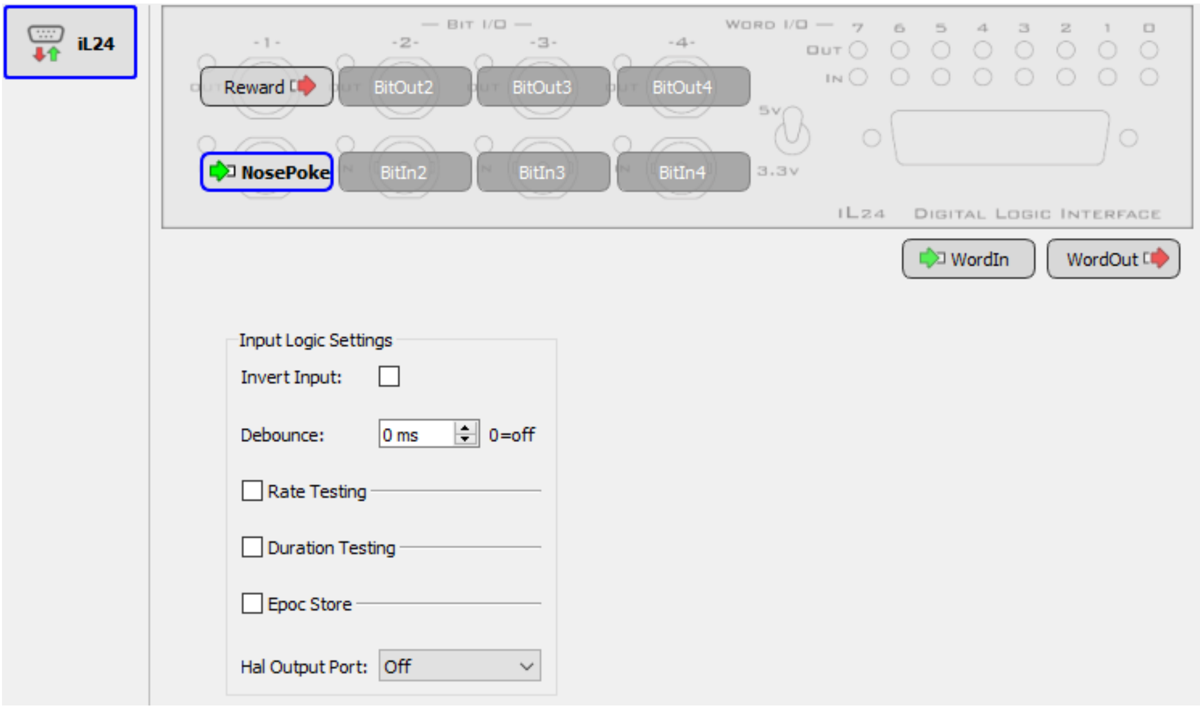
See the Hardware Manual for information on iL24 technical specifications.

iL24 Options



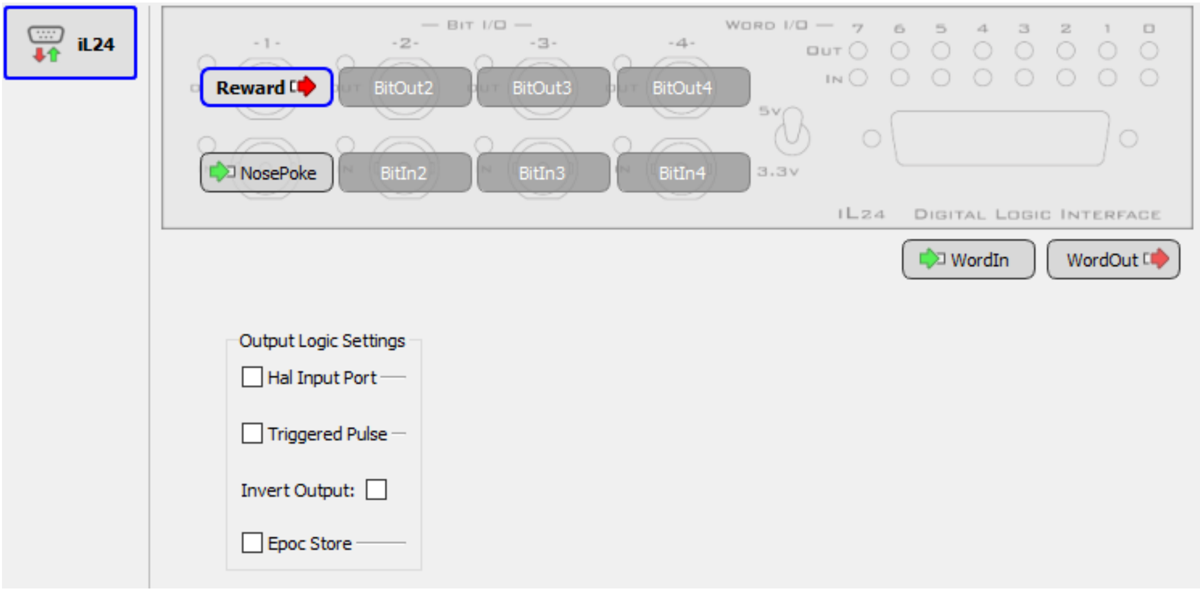
iL24 Interface

Click on a port to enable it. Set the **Name** of the port to something that makes sense for your experiment, e.g. 'NosePoke'.



iL24 Input Options

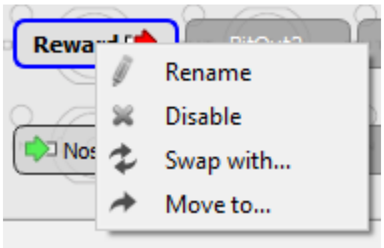
All iL24 inputs go through a logical input processor. See Logic Input Processor for more details.



iL24 Output Options

All iL24 outputs go through a logical output processor. See Logic Output Processor for more details.

After you define a port, you can modify it or move it by right-clicking on the port.



iL24 Context Menu

WordIn / WordOut

All 12 input bits can be read from the WordIn port. The lowest four bits are the BNC inputs, and the DB25 inputs are shifted into the upper 8 bits.

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	N/A	N/A	N/A	N/A	BI4	BI3	BI2	BI1

All 12 output bits can be controlled from the WordOut port. The lowest four bits are the BNC outputs, and the DB25 outputs are shifted into the upper 8 bits.

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0	N/A	N/A	N/A	N/A	BO4	BO3	BO2	BO1

See the iL24 Pinout for physical pin locations on the DB25. For more information on using digital inputs and outputs, see Understanding Digital I/O User Guide.

interfaces with Feed3, bpod, arduino, any TTL based system

iMn Multi-Function Interface

The iM5, iM7, iM8, iM9, iM14, and iM15 have specialized I/O functions, including advanced audio processing critical for auditory behavioral experiments.

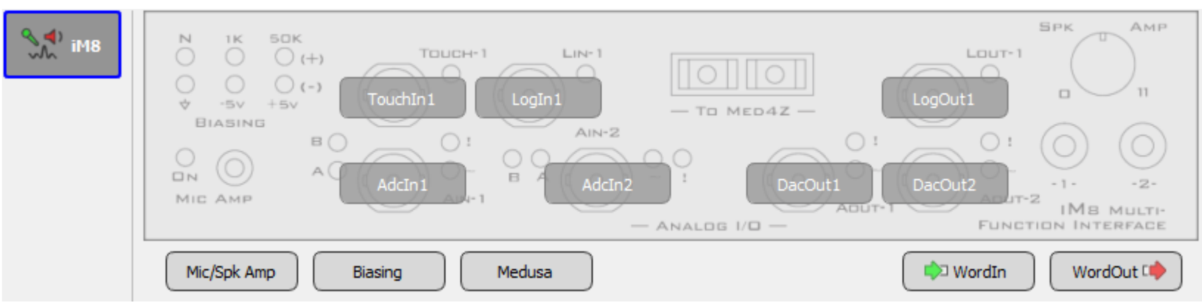
- Drive a speaker (e.g. MF1) directly up to 4W via an RCA connection.
- Record cage sounds via an embedded microphone amplifier.
- Connect with many common cage elements (e.g. touch sensors) via BNC ADC inputs and DAC outputs.
- Generate AM and FM stimulation (excludes iM10)

	Analog In	Analog Out	Digital In	Digital Out	Touch Inputs	Accessory Port
iM5	1	1	1	1	-	-
iM7	1	1	1	1	-	Medusa Amp
iM8	2	2	1	1	1	Medusa Amp
iM9	2	2	2	2	1	-
iM14	1	1	1	1	-	iH8 +28 V*
iM15	2	2	2	2	1	iH8 +28 V*

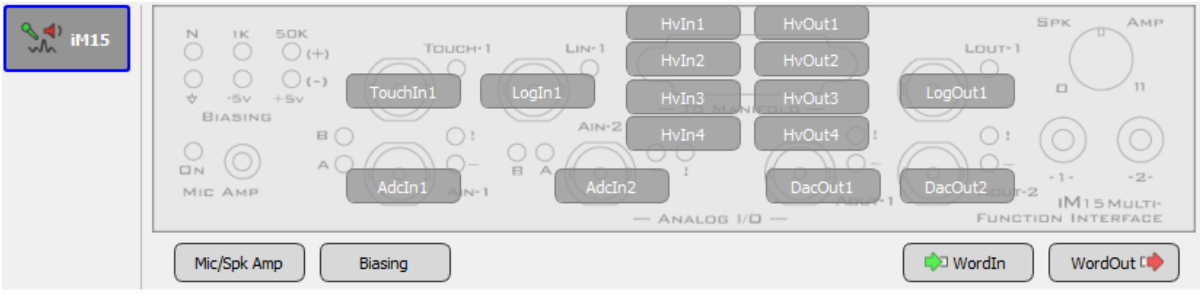
* The +28 V logic includes four output bits and four input bits.

See the Hardware Manual for information on iMn technical specifications.

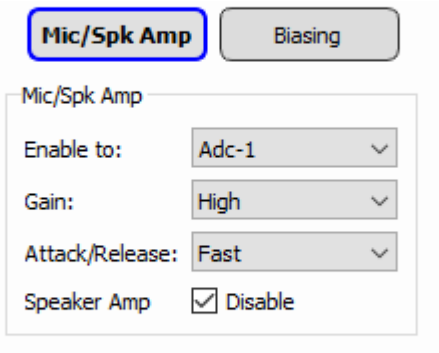
iMn Options



iM8 Interface



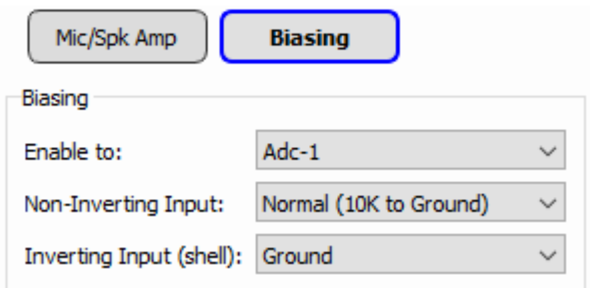
iM15 Interface



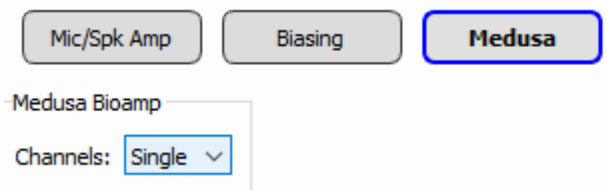
The **Mic/Spk Amp** setting lets you optionally enable the ⅛" jack to use for Analog Inputs 1 and 2 (if available). It also adds 56.6 dB, 66.6 dB, or 76.6 dB gain to the input, and set Attack/Release ratios of 1:500, 1:2000, or 1:4000.

Important

Do not connect anything to the BNC ADC input(s) when the microphone amp is enabled.



The **Biasing** setting lets you optionally adjust the input impedance and bias voltage on Analog Inputs 1 and 2 (if available).



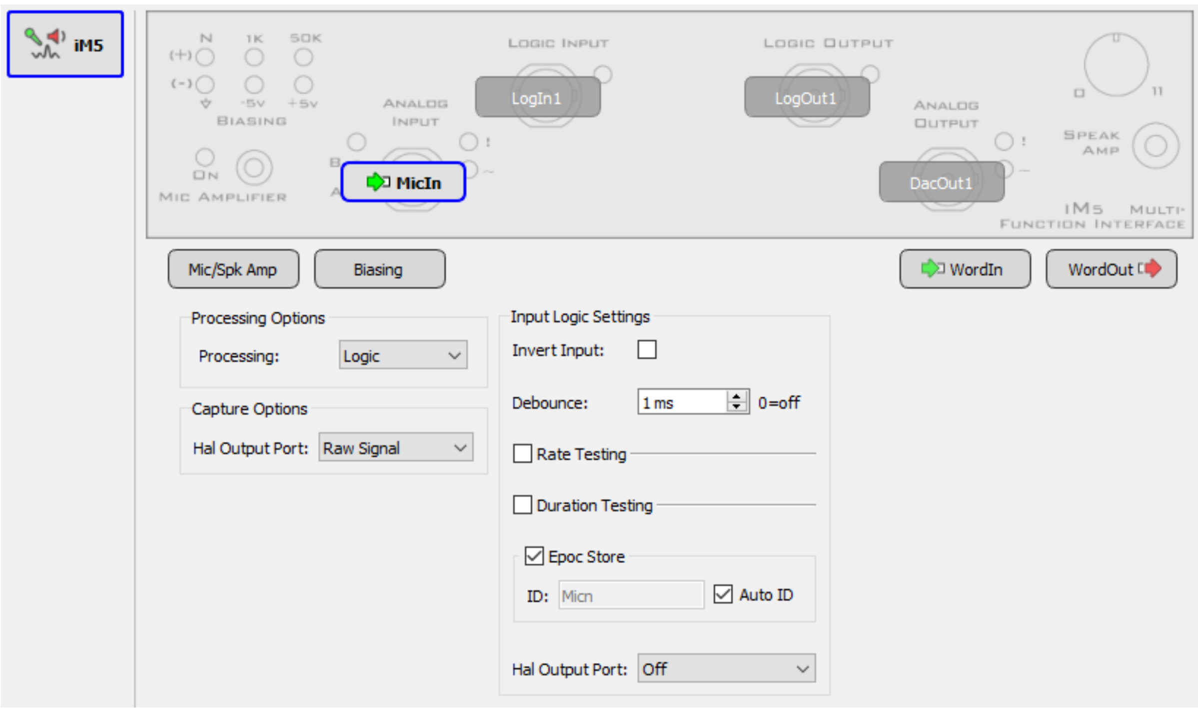
The **Medusa** setting (iM7 and iM8 only) sets the number of channels to read from the Medusa preamplifier port (0-4).

The iM14 and iM15 modules have a DB15 port that connects to an iHm manifold. There are four +28 V outputs (HvOut1-HvOut4) that map to 1-4 on the manifold and four +28 V inputs (HvIn1-HvIn4) that map to 5-8 on the manifold.

All iMn inputs go through an input processor. Analog signals are first converted into TTL logic signals using the methods described below, and then pass through a logic processor. See iMn Input Processor for more details and a diagram of the complete process for the iMn analog inputs.

Analog Inputs

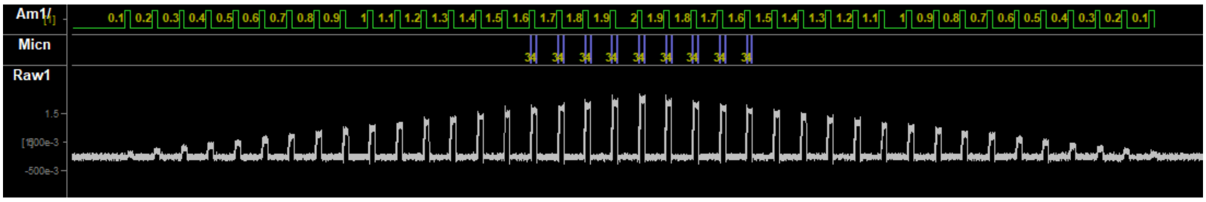
Logic



iM5 Logic Input Options

Input signal above ~1.5 V triggers the logic input.

Example



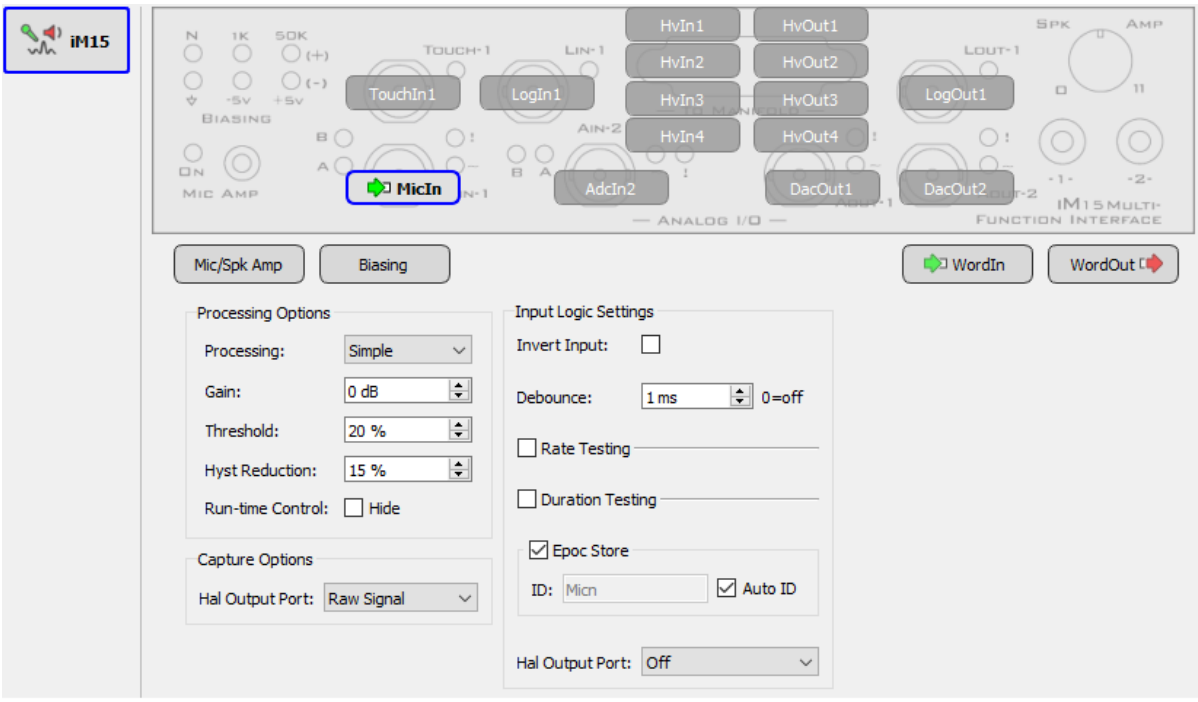
iM5 Raw Input Signal and Logic Threshold Crossing

Signals above ~1.5 V trigger the `Micn` epoc store. As with all iCon input epoc events, a value of 3 is saved on the `Rising` edge and a value of 4 is saved on the `Falling` edge.

The rest of the processing is handled by the Logic Input Processor.

Simple

Use the Simple option if you want to add gain to the analog input signal and manually set a threshold.



iM15 Simple Input Options

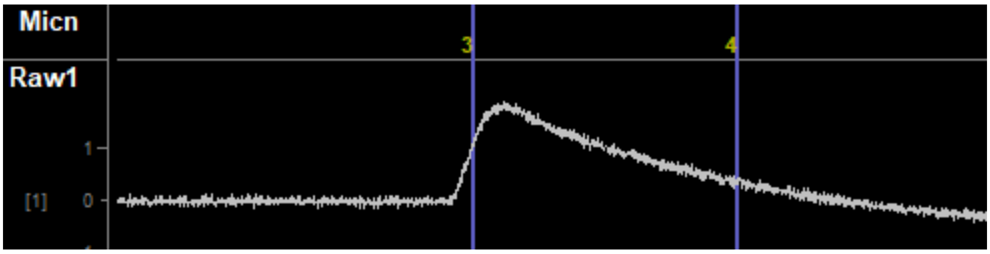
Option	Description
Gain	Apply up to 76.6 dB gain
Thresh	Set logic threshold as a percentage of the 5 V input range, after gain is applied. This can be negative
Hyst Reduction	Hysteresis Reduction. Set a percentage away from the Thresh the signal has to cross to turn the input off

Note

These settings are adjustable in the run-time interface

If threshold is positive, the lower threshold is (Thresh % - Hyst %).

Example

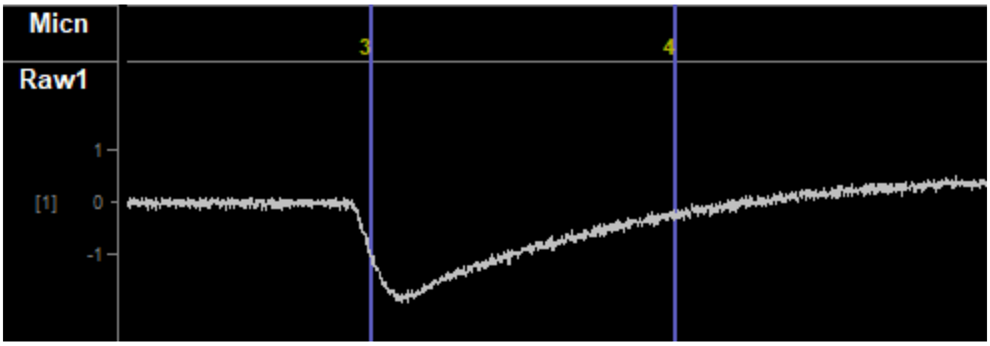


iM15 Raw Input Signal and Simple Threshold Crossing

Thresh is 20% and Hyst is 15%. The input logic is true when the signal crosses above 20% of the input range (20% of 5 V = 1 V) and false when the signal goes below 5% of the input range (0.25 V).

If threshold is negative, the lower threshold is (Thresh + Hyst).

Example



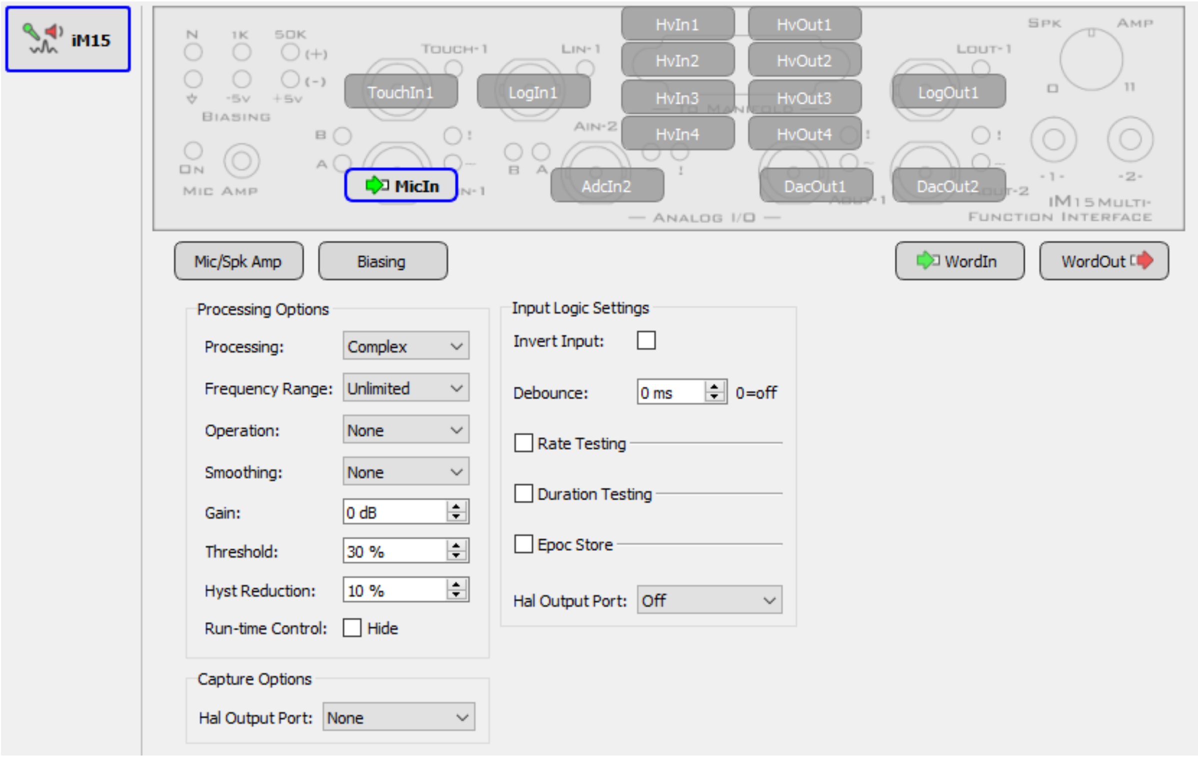
iM15 Raw Input Signal and Simple Threshold Crossing

Thresh is -20% and Hyst is 15%. The input logic is true when the signal crosses below -20% of the input range (-20% of 5 V = -1 V) and false when the signal goes above -5% of the input range (-0.25 V).

The rest of the processing is handled by the Logic Input Processor.

Complex

The iMn analog inputs run at 400 kHz, independent of the RZ / iConZ sampling rate. Use the Complex options to significantly increase fidelity in the frequency range you are interested in by reducing the input bandwidth. You can also apply non-linear operations such as absolute value, square, flip the signal sign, or add smoothing.



iM15 Complex Input Options

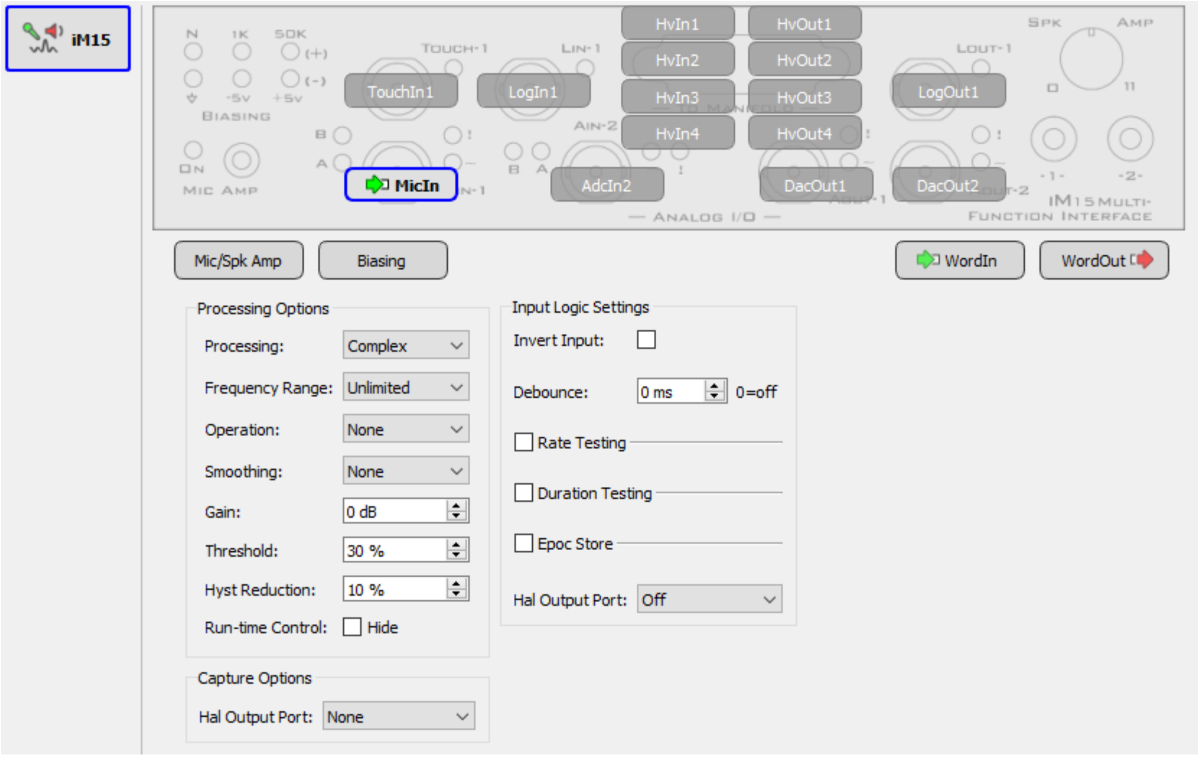
Option	Description
Frequency Range	Limit the frequency range of the input
Highpass Freq	Set highpass filter frequency (if `Frequency Range` is enabled)
Lowpass Freq	Set lowpass filter frequency (if `Frequency Range` is enabled)
Operation	Take the Absolute Value, Square the signal, or Flip the sign on the input
Smoothing	Set the time constant for exponential smoothing of the input signal

> **Note**
>
> All settings except for `Frequency Range` and `Operation` are adjustable in the run-time interface.

See Simple processing options for a description of the rest of the settings and thresholding examples.

The rest of the processing is handled by the Logic Input Processor.

Clip Detect



iM15 Clip Detect Input Options

Triggers when signal is within ~3% of the 5 V input range (signal is greater than approximately ±4.8 V).

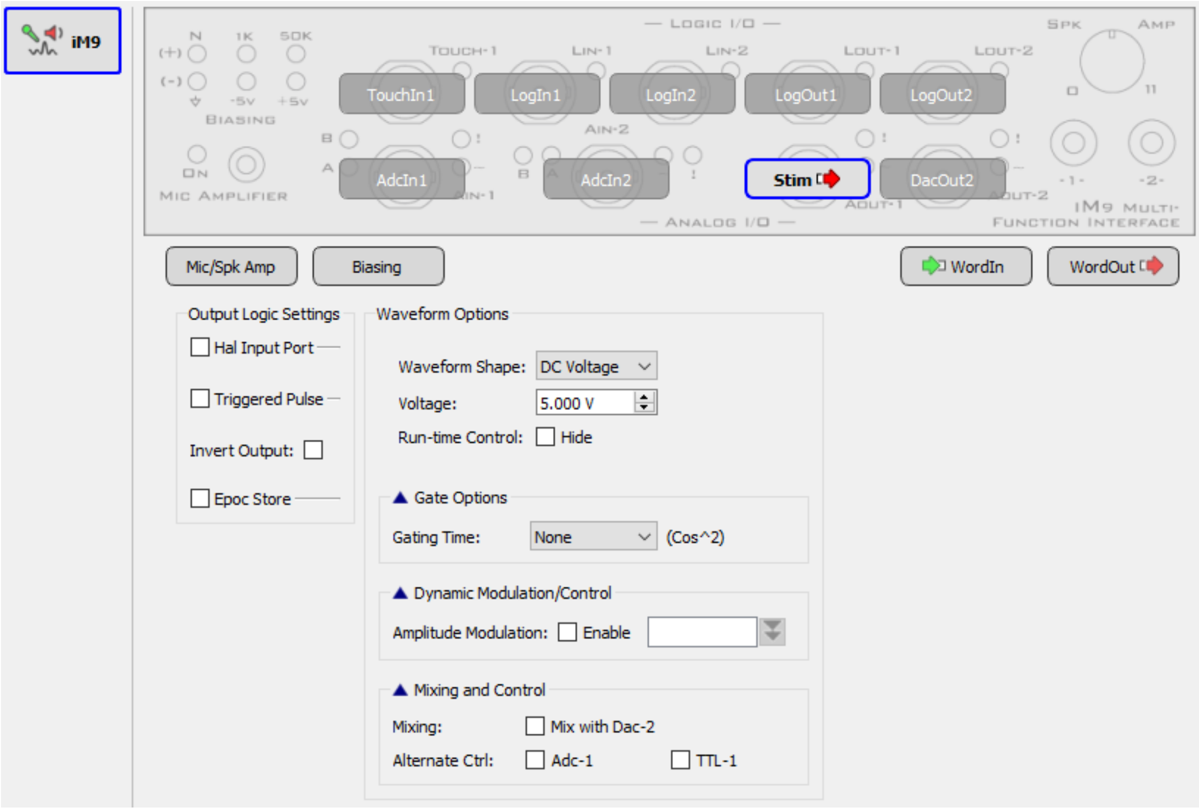
The rest of the processing is handled by the Logic Input Processor.

Touch Inputs

Touch sensor inputs are available on the iM8, iM9, and iM15 only. All touch sensor inputs are handled by the Logic Input Processor.

Analog Outputs

All iMn outputs first go through a logical output processor that controls when the output is presented. See Logic Output Processor for more details.



iM9 Output Options

Waveform Shape	Description
User	Send an output signal from another gizmo's output link to the iMn DAC
DC Voltage	Output a DC voltage when triggered
Tone	Generate up to 100 kHz tone **^**
White Noise	Generate a band-limited white noise
Pink Noise	Generate pink noise (1/f)
Square	Generate up to 100 kHz square wave (50% duty cycle, ±5 V) **^**
Clock	Generate up to 100 kHz clock signal (50% duty cycle, +5 V) **+**
PWM	Generate a pulse-width modulated waveform using 800 kHz clock (1.25 us minimum pulse width)

Option	Waveform Shape	Description
Gating	All	Add an optional cos2 gate to the waveform
Phase	Tone, White Noise, Pink Noise, Square, and Clock only	When `Frozen` is checked, the phase of the waveform resets to 0 when triggered
Freq Resolution	Tone, Square, and Clock only	Adjusting the frequency resolution changes the maximum frequency that the iMn can generate **^**
Base Frequency	PWM only	Frequency of the PWM square waveform onsets
Duty Cycle	PWM only	Duration of the PWM `on` time, as a percentage of the base period
Attenuation	Tone, White Noise, Pink Noise, Square, Clock, and PWM only	iMn-generated waveforms use a 5 V amplitude. Apply up to 50 dB attenuation to the signal, adjustable at run-time.

Modulation	Waveform Shape	Description
Amplitude	User, DC, Tone, White Noise, Pink Noise, Square, Clock	Control the waveform amplitude dynamically from another gizmo. The signals are multiplied, so typically 0 to 1
Frequency	Tone, Square, Clock	Control the waveform frequency dynamically from another gizmo. The Freq Resolution still applies - the incoming frequency control signal is rounded to the nearest allowed frequency
Duty Cycle	PWM	Control the duty cycle percentage dynamically from another gizmo, 0 to 1 (1 = 100%)

^ For audio applications, set Freq Resolution to 10 Hz or 100 Hz only for best results.

+ For precision control applications, use PWM instead

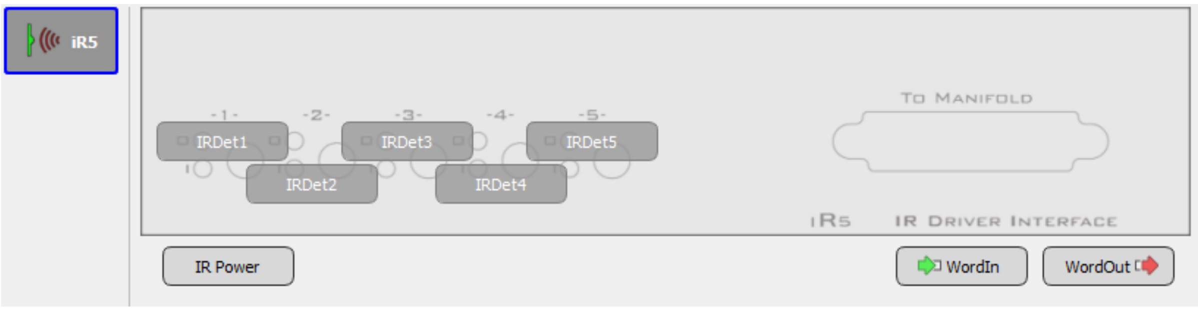
iRn IR Driver Interface

The iR5 and iR10 are specialized interfaces for infrared sensor beams. These modules have built-in power and logical connections to drive external IR sensors and send TTL events whenever the subject crosses the beam, without the need for any external signal processing.

- Manual trigger buttons for each IR port simplify testing and debugging
- Status lights for each IR beam monitor subject movement during sessions.

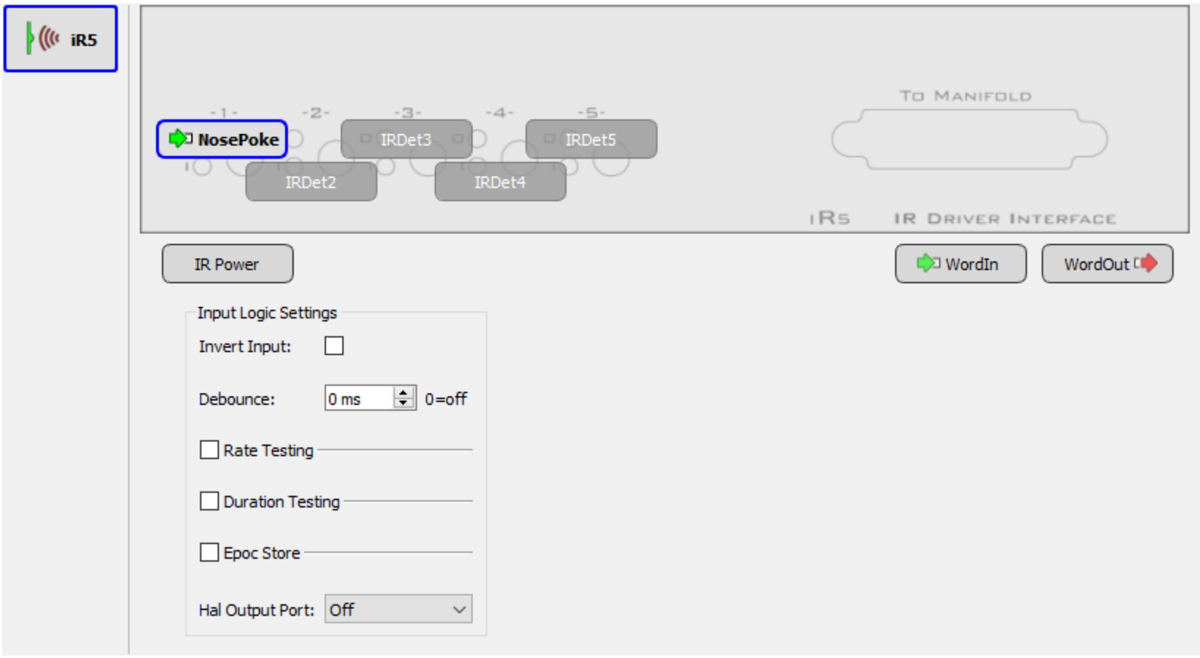
See the Hardware Manual for information on iRn technical specifications.

iRn Options



iRn Interface

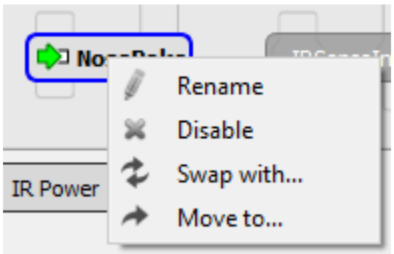
Click on a port to enable it. Set the **Name** of the port to something that makes sense for your experiment, e.g. 'NosePoke'.



iRn Input Options

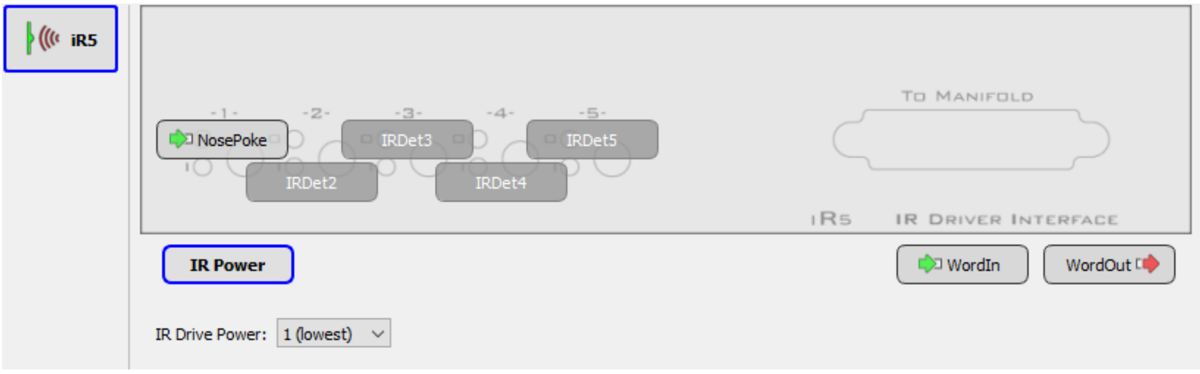
All iRn inputs go through a logical input processor. See Logic Input Processor for more details.

After you define a port, you can modify it or move it by right-clicking on the port.



iRn Context Menu

The **IR Power** setting lets you set the IR Drive Power (1 to 8).



iRn Power Setting

iS9 Aversive Stimulator

The iS9 generates small electrical currents for standard behavioral conditioning experiments, such as startle and Pavlovian fear response. Timing is precisely controlled for pairing the stim with other sensory cues or behavioral events. The iS9 is compatible with 3rd party behavioral boxes (Med Associates, Lafayette, and Harvard Apparatus).

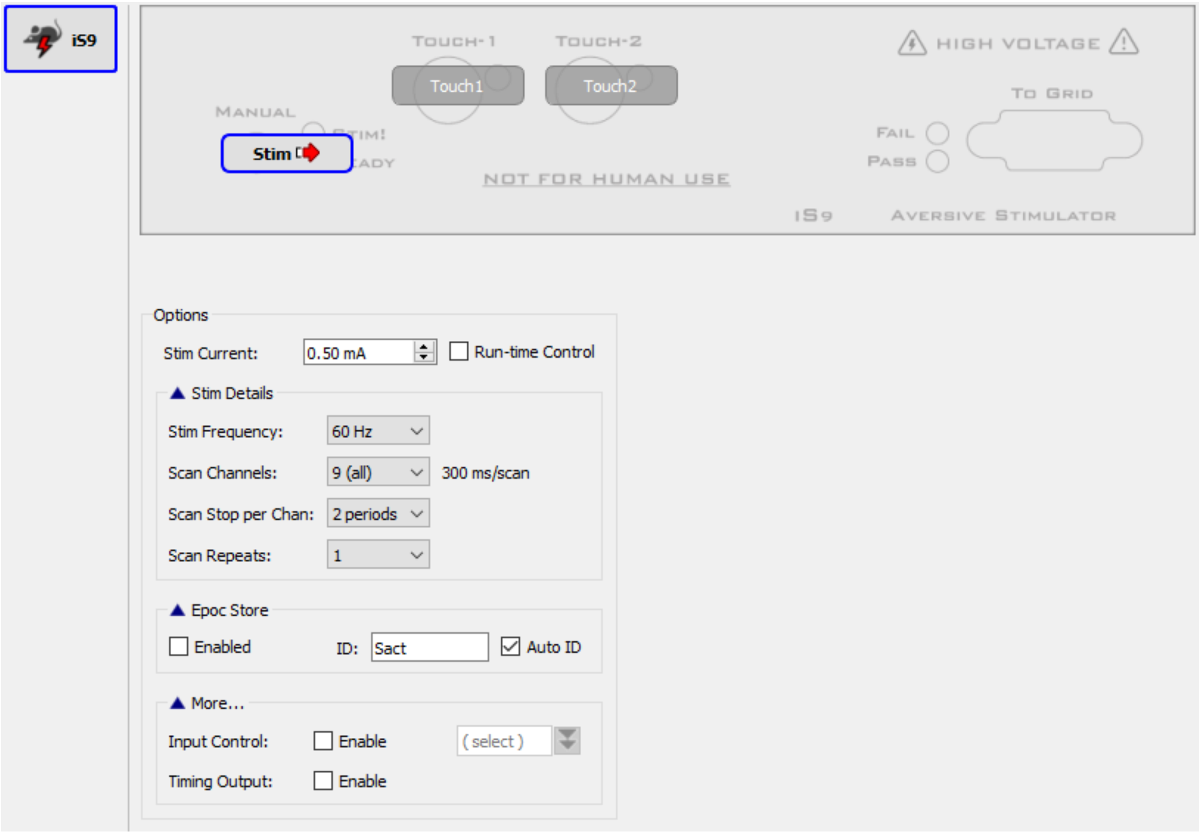
- Deliver current from 0 mA - 2.5 mA in 0.01 mA steps to each of 9 metal floor grids in rapid succession.
- Detect spout licks with built-in touch sensors (decoupled at stimulation to avoid false responses).
- Detect null events when stimuli fail to reach the subject.
- Automatically time-stamp stimulus and behavioral event information for on- and off-line analysis.

See the Hardware Manual for information on iS9 technical specifications.

iS9 Options

iS9 Interface

Stim Outputs



iS9 Output Options

Set the stim frequency, number of channels, stim periods per channel, and scan repeats

Touch Inputs

All touch sensor inputs are handled by the Logic Input Processor.

iS9 input Options

iVn Video Capture Interface

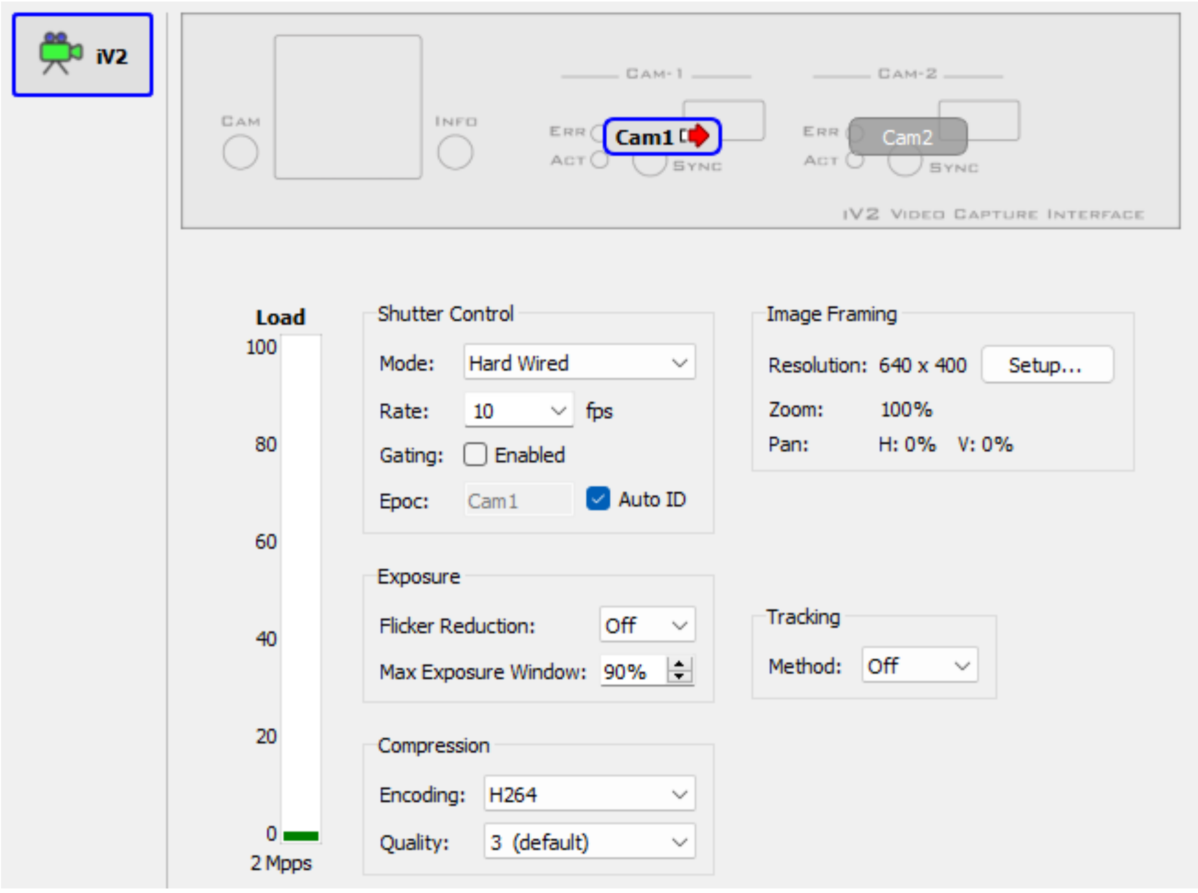
The iVn iCon module connects to high-resolution USB3 cameras to record video frames synchronized with all behavioral, electrophysiology, fiber photometry, and other research data. Record video from one camera with an iV1, or up to two cameras simultaneously per iV2 module. The iV2 module also includes real-time subject tracking.

- Records video from one or two USB cameras
- Synchronizes video frames with all other recorded behavioral and neurophysiological data
- Displays selected camera output on unit for easy camera setup and subject monitoring

Frame data is sent through the iCon interface back to the PC, shown on the Synapse window, and saved with the rest of your data as an MP4 file. You can review the video files along with your data in OpenScope.

See the Hardware Manual for information on iVn technical specifications.

iVn Options



iV2 Interface

Shutter Control

Mode	Description
Hard Wired	Uses the 'Sync' output on the iVn module to trigger the camera. Recommended option.
Soft Wired	Frame capture signal is sent from Synapse to iVn module's computer to trigger frame capture. Use when hardware sync cable is not available.
Free Running	The iVn module handles the frame timing. Not recommended.

Gating allows you to turn frame capture on / off dynamically, if you only want to capture frames around particular events. See Capture Control below.

Exposure

Setting	Description
Flicker Reduction	Changes the exposure to match the beat frequency of the lighting
Max Exposure Window	The iVn automatically adjusts the exposure setting at run-time. You can limit how long the exposure is (as a percentage of the frame capture window) to give the iVn more time to capture / return the frame.

Compression

Setting	Description
Encoding	Choose iVn onboard encoding algorithm. H264 (recommended), H265, or MJPEG.
Quality	This setting determines the overall data rate (bits per second) for the streaming video signal from the iCon back to the processor, regardless of resolution or FPS.

Image Framing

Tracking

The **Store to CSV** option saves a CSV file with the same prefix as the mp4 video file. The format of each row of the csv file is the frame number followed by information on each of the tracked points:

```
{FRAME_NUMBER},{TRACKED_POINT_1},{TRACKED_POINT_2},...,{TRACKED_POINT_N}
where {TRACKED_POINT_N} is:
{POINT_ID_N},{IS_VALID_N},{CONFIDENCE_N},{REGION_N},{X_N},{Y_N}
```

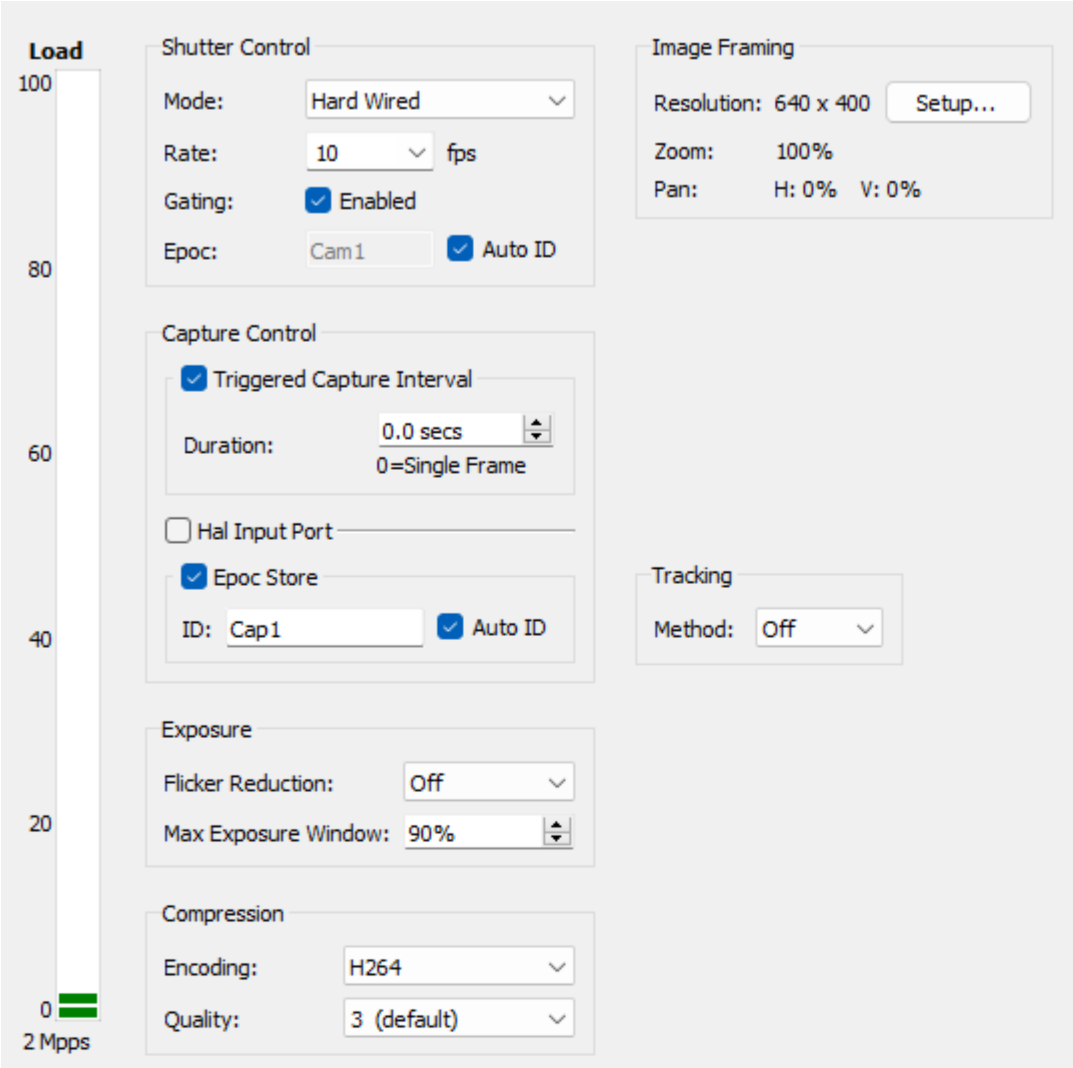
Load Management

At Idle time, the Load meter dynamically updates the processing load on the iVn. This calculation is based on the type of module (iV1 vs iV2), the number of cameras, the requested frame rate and resolution, the encoding for each camera, and the complexity of the tracking algorithm (iV2 only). This gives you an idea of the load on the system. Always double-check for lost frames in the Synapse iCon tab at run-time.

Image dynamics and how well it can compress the images. A rapidly changing image will consume more iVn processing power and be lower quality to meet the compression bandwidth setting limits. The iVn has an onboard 10 second frame buffer that allows it to catch up after periods of rapidly changing images. A small progress bar indicator on the LCD screen shows the state of this buffer. If it fills up, you will see Frame Errors in the Synapse run-time tab, indicating lost frames. The frame buffer is shared by both cameras.

Capture Control

The Capture Control options appear when Gating is enabled.



iV2 Gating Options

Triggered Capture Interval enables frame capture for a specified amount of time when triggered. This can be triggered manually through a button on the run-time iCon tab interface, from another gizmo (Hal Input Port), or through the SynapseAPI.

Synapse API Control

If using Triggered Capture Interval, you just need to trigger frame acquisition with one call:

```
import tdt
syn = tdt.SynapseAPI()
syn.setParameterValue('iCon(1)', 'V1_OutEnab_1', 1)
```

Otherwise you can toggle capture on/off with this:

```
import time
import tdt
syn = tdt.SynapseAPI()
syn.setParameterValue('iCon(1)', 'V1_OutEnab_1', 1)
time.sleep(1)
syn.setParameterValue('iCon(1)', 'V1_OutEnab_1', 0)
```

If Gating is disabled, or if Gating is enabled and Hal Input Port is True, you can use `FrameEnab` to selectively override frame capture:

```
import time
import tdt
syn = tdt.SynapseAPI()
syn.setParameterValue('iCon(1)','V1_FrameEnab',1)
time.sleep(1)
syn.setParameterValue('iCon(1)','V1_FrameEnab',0)
```

Getting Started

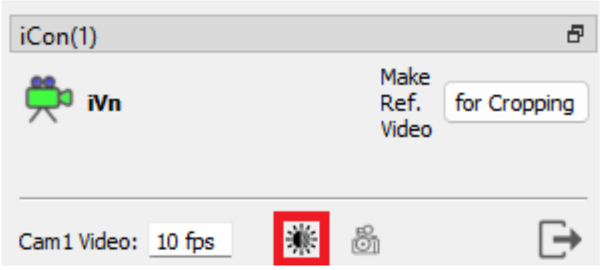
Start by clicking the "Cam1" button to enable the first camera (and "Cam2" if you have a second camera) and Commit.

Reference Video

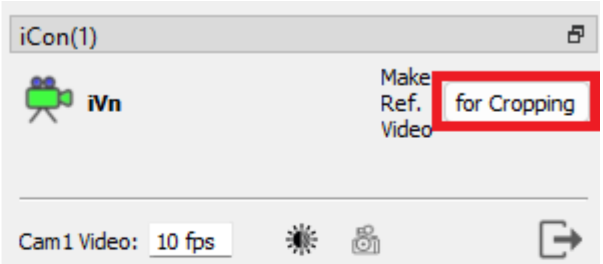
The first thing to do is create a reference video. This gives you a frame in Synapse to set up area of interest for recording. It also retrieves the camera properties which includes the available resolutions.

To create a reference video:

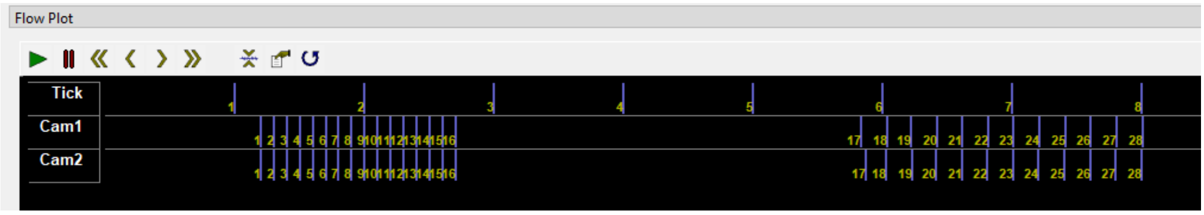
1. Click "Preview" in Synapse.
2. Position the camera to capture the area of interest. Use the front panel of the iVn module or the iCon tab in Synapse run-time window to view the camera image.
3. On the iCon tab in Synapse run-time window, click the small brightness/contrast icon to open the camera hardware settings to adjust brightness, contrast, etc.



4. Click the "Make Ref. Video for Cropping" button.



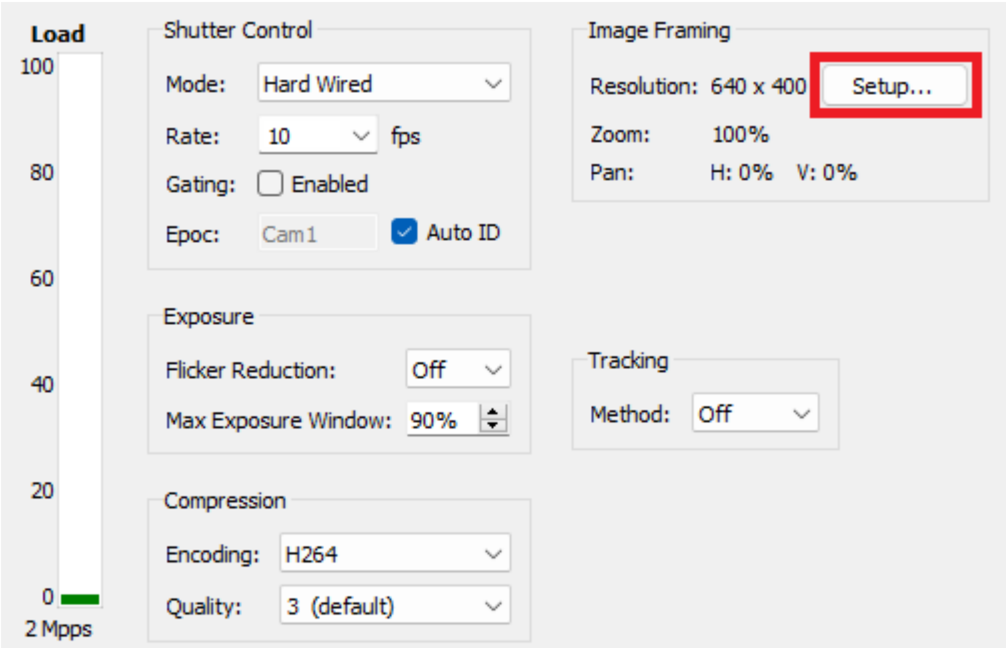
5. Wait ~10 seconds and verify that the image is updating and you see Cam epocs. There will be a gap in the epoc events as it switches to Reference Mode, as shown in the example below:



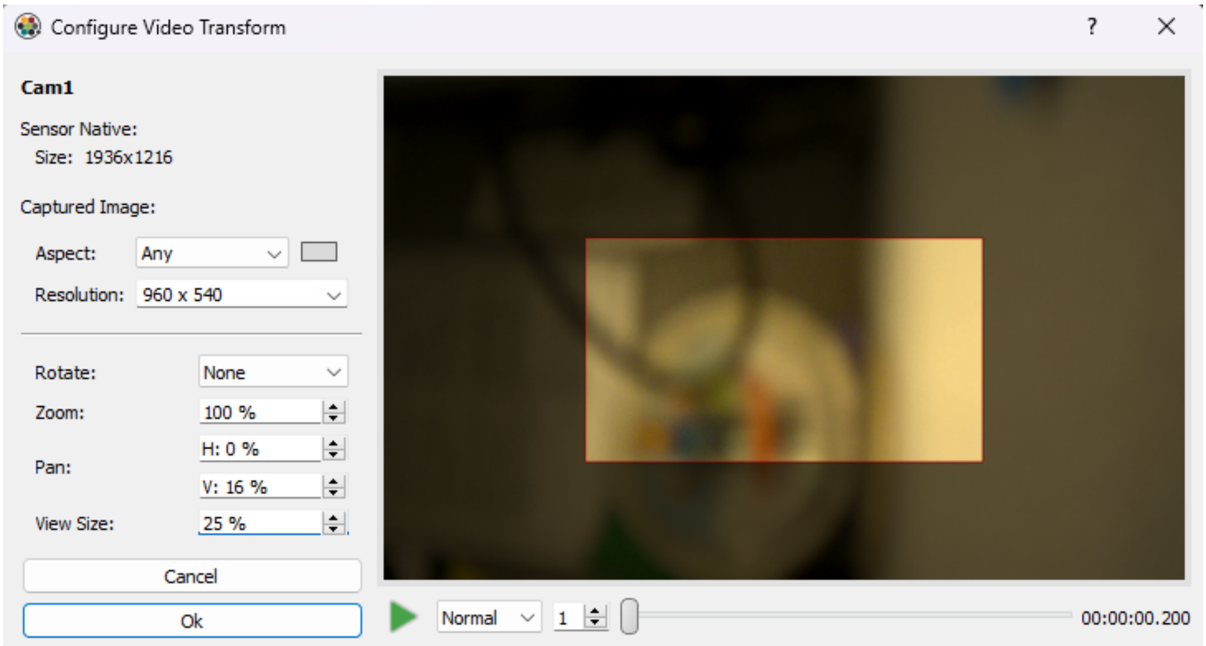
6. Click the Synapse "Idle" button to return to the iVn configuration. Synapse saved a short video at 5 fps containing the full sensor image, and retrieved the available camera resolution settings.

> **Note**
>
> The reference videos are saved in `C:\TDT\Synapse\CamRefs` with the prefix `FrameRef`

7. In Idle mode, click the "Setup" button.



8. This opens the "Configure Video Transform" dialog. This is where you set the desired camera resolution, and set rotation and digital zoom / cropping on the image.



9. Click "OK" to exit the dialog, and click "Commit" in Synapse to save your experiment changes.

10. If you have a second camera, repeat Steps 7-9 for "Cam2".

Real-Time Tracking

The iV2 module offers two methods for computing real-time positional information of the subject. **R-Track** is a rodent tracking method of searching for the body and head of a rodent. **D-Track** is a dot tracking method that searches for the location of a certain color within the frame.

Contact TDT Support support@tdt.com for help setting up real-time subject tracking.

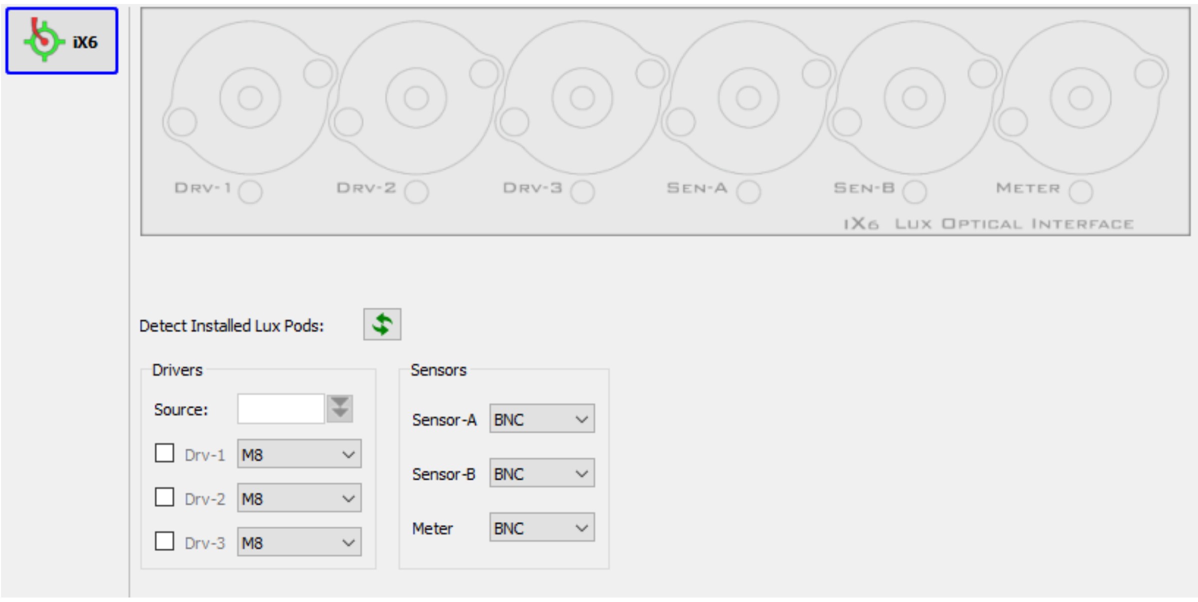
iXn Lux Optical Interface

The iX6 iCon module drives integrated LEDs and records fluorescent responses for fiber photometry and optogenetics applications. Each module can be configured with up to three LEDs from a wide selection of wavelengths.

- Drives up to three integrated LED outputs of selectable wavelengths from TDT's Lux product line.
- Measures and records fluorescent responses via two integrated photosensor inputs.
- Monitors light power output at the subject level with an integrated photo meter.

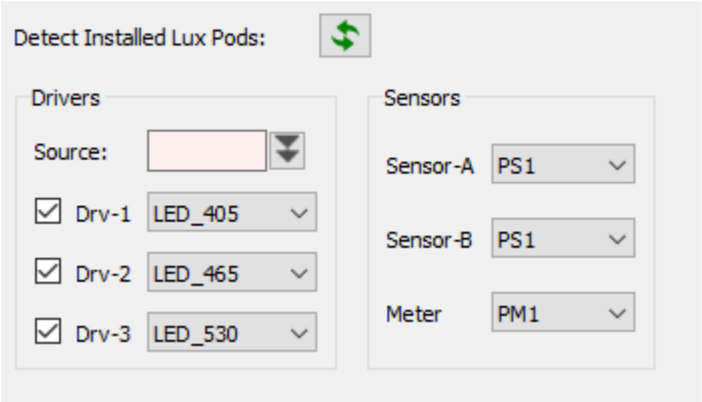
See the Hardware Manual for information on iX6 technical specifications.

iXn Options



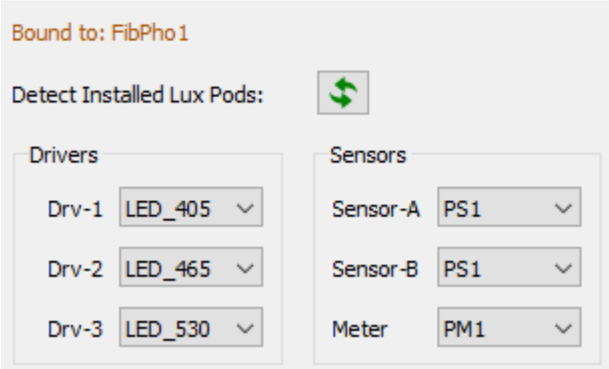
iX6 Interface

Start by clicking the cycle button to automatically detect the Lux Pods.



iX6 After Detect

Then attach a Fiber Photometry gizmo to the RZ in your Processing Tree. The iX6 configuration will propagate to the Fiber Photometry gizmo that it attaches to.



iX6 Attached to Fiber Photometry Gizmo

See the Fiber Photometry User Guide for more usage information.

BH32 Behavioral Controller Interface

Important

The BH32 is deprecated, consider the iCon Behavioral Interface as a replacement for control of behavioral cage elements.

The BH32 interface can send and/or receive bit-wise or byte-wise UDP packets to the BH32 Behavioral Cage Controller. The physical RZ processor must have an Ethernet port on the back (labeled "UDP") and the BH32 must be on the same network as the RZ processor.

Adding BH32 to your Rig and Processing Tree

The BH32 isn't added to the rig automatically.

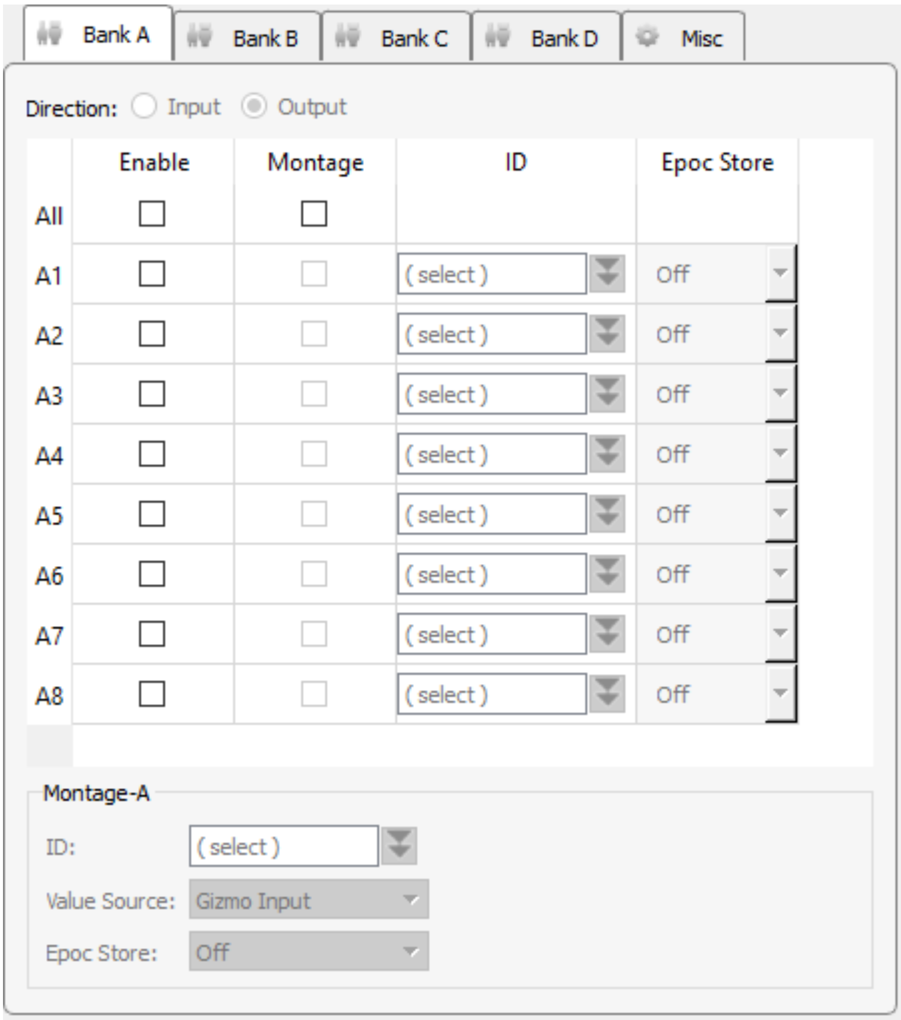
To add a BH32:

1. Click **Menu**, then **Edit Rig**.
2. In the Rig Editor, right-click your system's RZ processor, then click **Add BH32**.
3. Click **OK** to close the Rig Editor and update the Processing Tree.

BH32

The BH32 communicates with the RZ processor through the Ethernet port. It sends and receives UDP packets out the RZ to external devices.

BH32 Options



BH32 Options

All bits are controlled through the Bank A, B, C, D tabs.

All of the output bits are controlled by either a Synapse signal link or via the API Access. Whenever the input signals change, a new packet is sent to the BH32.

All of the input bits are available as an output links in Synapse that can be connected to other gizmos.

There is a NewPacket TTL signal that strobes when new data is received.

BH32 Misc Tab

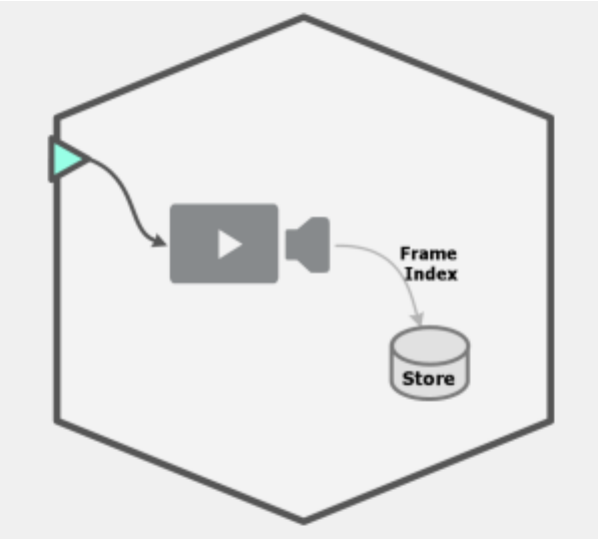
Select the **Save Raw Word** check box to store the received packet in the data tank as a timestamped event.

See the BH32 section of the System 3 Manual for more information about BH32 setup and operation.

CamHal Software USB Camera

> **Important**
>
> The Cam Hal is deprecated, consider the iV1 module as a replacement for precisely-timed low-bandwidth frame acquisition.

> **Important**
>
> The CamHal is not compatible with the iPac and SynCon software.



Camera Block Diagram

The Camera HAL captures images from a USB camera directly connected to your computer for general subject behavior monitoring. Frames are captured, stored to disk and timestamped in the data block. Camera frames are saved as a DIVX-encoded AVI file in the same folder as your data block, in the form `{TANK}_{BLOCK}_{HalName}.avi`. The frame numbers are stored in the data tank as epoch events. The AVI file can be used with the OpenScope Video Viewer and annotation tools.

Only USB cameras that support OpenCV direct operating system camera drivers can be used with the Camera HAL in Synapse. Many industrial cameras (e.g. Basler) don't provide standard driver interfaces for the operating system.

Synapse supports up to two cameras per Rig and frame rates up to 20 fps. For best practice when using two cameras, keep cameras on separate USB Buses. Typically, PCs use separate buses for rear and front accessible USB inputs.

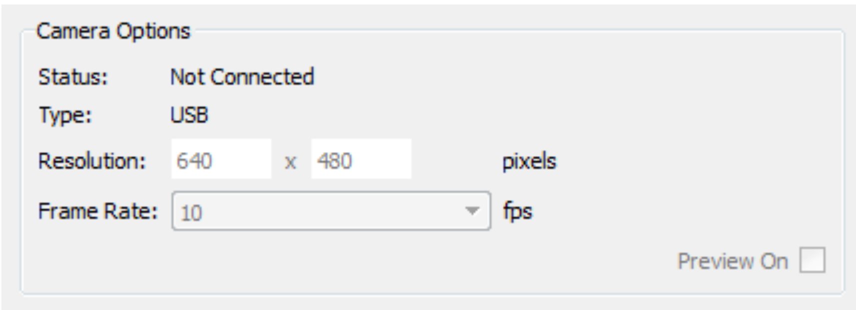
> **Important**
>
> Frame rate is controlled by a software timer and is prone to jitter. For real-time synchronized video capture, use the iV1 module.

Adding the HAL

Cameras are not automatically detected in the Synapse Rig Editor and must be manually added. To add a camera:

1. Click **Menu**, then **Edit Rig**.
2. In the Rig Editor, Right-click your system's processor (such as RX or RZ devices), then click **Add Cam**.
3. Click **OK** to close the Rig Editor and update the Processing Tree.

Camera Options



Camera Options

Option	Description
Status	Shows if camera is connected or not
Resolution	Select a video resolution
Frame Rate	Select desired capture rate in fps
Preview On	Shows a live stream from the camera in the Options area. The preview can be used to verify correct Camera placement and connection before recording.

Runtime Interface

The runtime tab displays the raw camera video for online monitoring. Please note that high-demand user interface tasks, like resizing windows, can increase jitter.

Runtime Plot with Camera Set to 5 Frames per Second

A subplot is also added to the main runtime multiplot to preview the frame index alongside other plot data.

Improving Performance

If you are having problems with things like dropped frames, we recommend installing the camera drivers to get access to more settings and controls for your camera. Some features like auto-adjust and "RightLight" can cause problems like increased jitter and dropped frames.

iM10 Multi-Function Interface

> **Important**
>
> The iM10 is deprecated, consider the other models in the iMn Multi-Function Interface line of products.

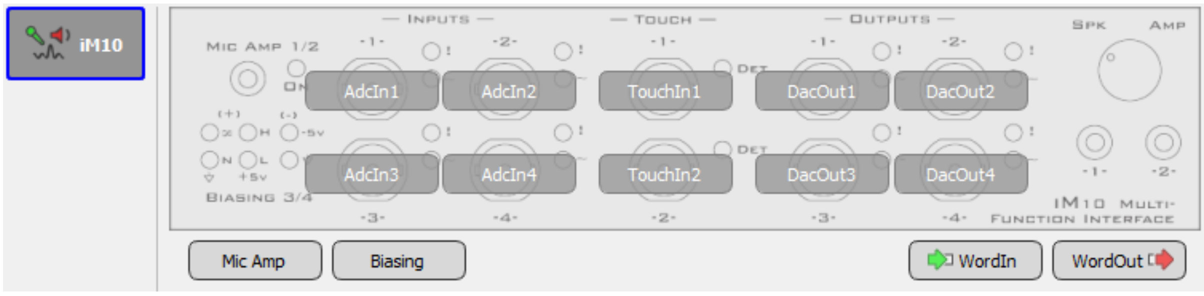
The iM10 has multiple specialized I/O functions, including advanced audio processing critical for auditory behavioral experiments.

- Drive a speaker (e.g. MF1) directly up to 4W via an RCA connection.
- Record cage sounds via an embedded microphone amplifier.
- Connect with many common cage elements (e.g. touch sensors) via BNC ADC inputs and DAC outputs.
- Streamline setup with a range of embedded signal processing options.

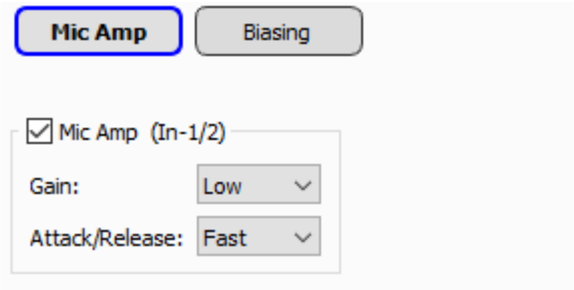
	Analog In	Analog Out	Digital In	Digital Out	Touch Inputs	Accessory Port
iM10	4	4	-	-	2	-

See the Hardware Manual for information on iM10 technical specifications.

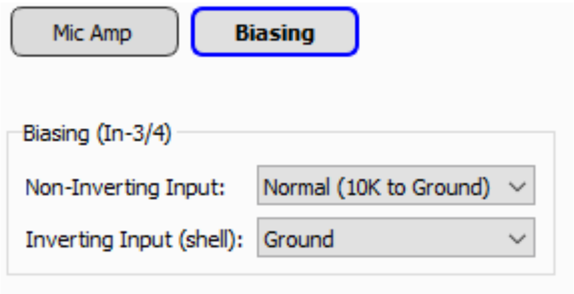
iM10 Options



iM10 Interface



The **Mic Amp** setting lets you optionally enable the ⅛" jack to use for Inputs 1 and 2. It also adds 56.6 dB, 66.6 dB, or 76.6 dB gain to the input, and set Attack/Release ratios of 1:500, 1:2000, or 1:4000.



The **Biasing** setting lets you optionally adjust the input impedance and bias voltage on Inputs 3 and 4.

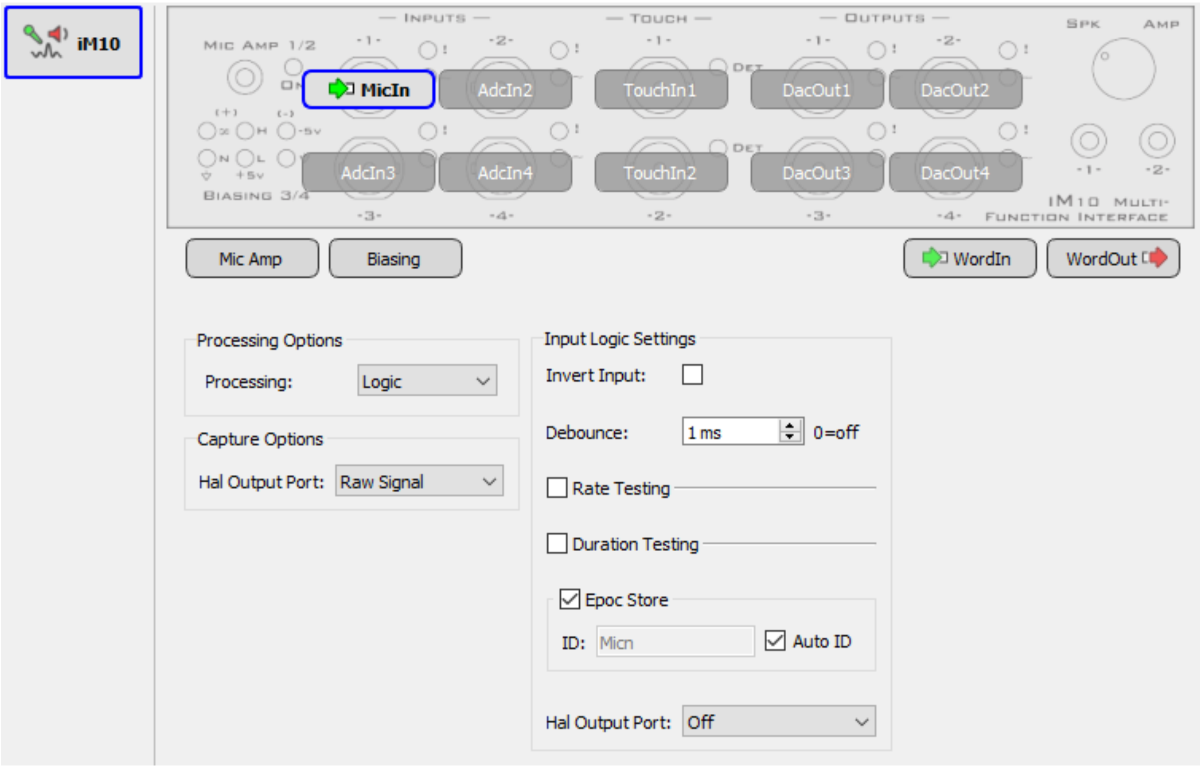
Important

Do not connect anything to the BNC Inputs 1 and 2 when the microphone amp is enabled.

All iM10 inputs go through an input processor. This first converts the analog signal into a TTL logic signal using the methods described below, and then goes through a logic processor. See iM10 Input Processor for more details and a diagram of the complete process for the iM10 analog inputs.

Analog Inputs

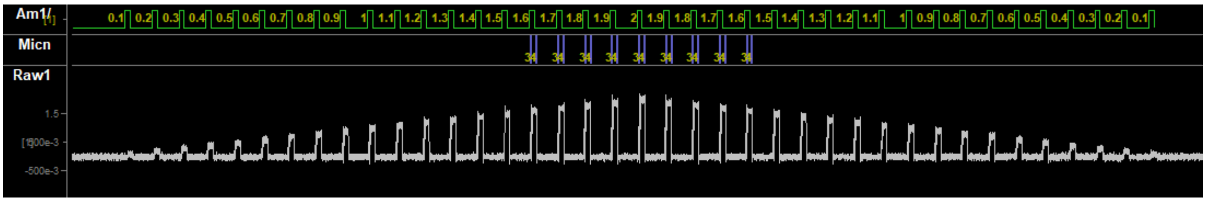
Logic



iM10 Logic Input Options

Input signal above ~1.5 V triggers the logic input.

Example



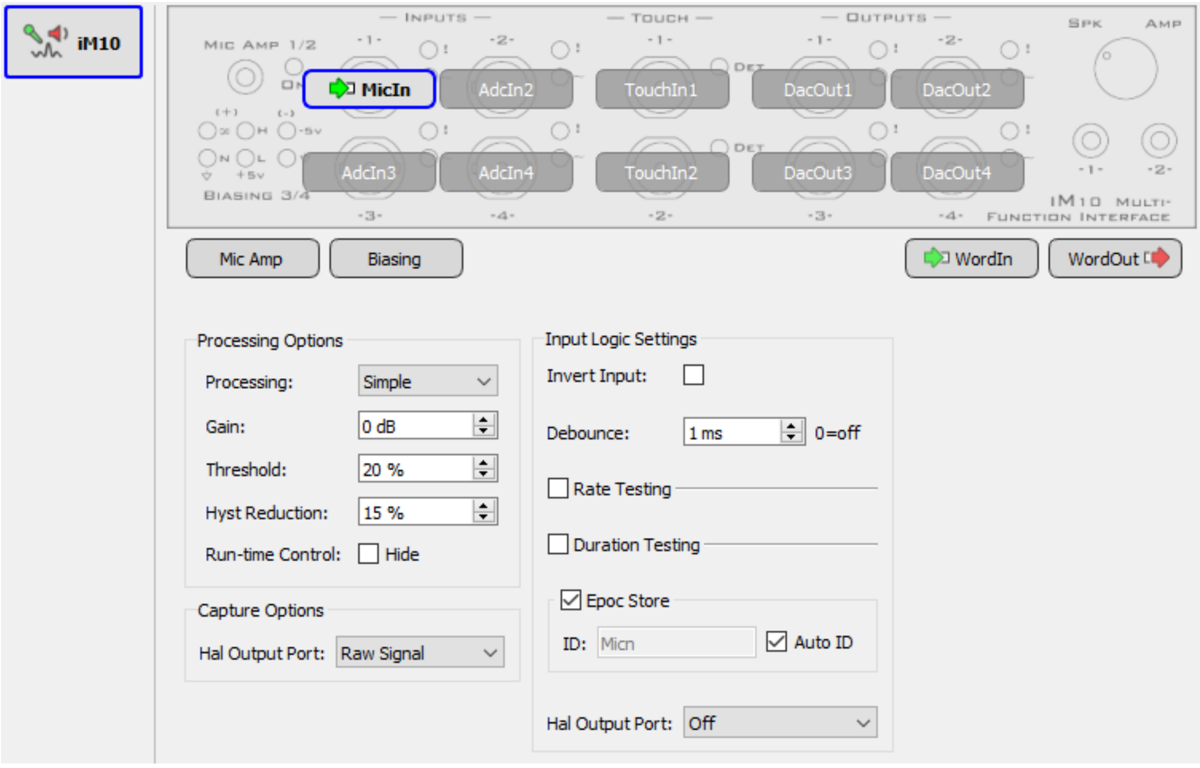
iM10 Raw Input Signal and Logic Threshold Crossing

Signals above ~1.5 V trigger the `Micn` epoc store. As with all iCon input epoc events, a value of 3 is saved on the `Rising` edge and a value of 4 is saved on the `Falling` edge.

The rest of the processing is handled by the Logic Input Processor.

Simple

Use the Simple option if you want to add gain to the analog input signal and manually set a threshold.



iM10 Simple Input Options

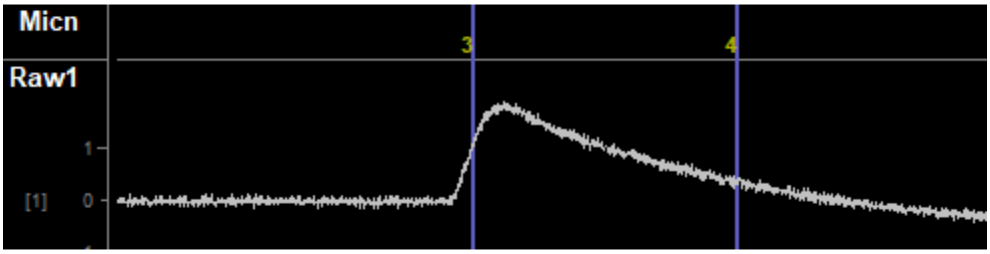
Option	Description
Gain	Apply up to 76.6 dB gain
Thresh	Set logic threshold as a percentage of the 5 V input range, after gain is applied. This can be negative
Hyst Reduction	Hysteresis Reduction. Set a percentage away from the Thresh the signal has to cross to turn the input off

Note

These settings are adjustable in the run-time interface

If threshold is positive, the lower threshold is (Thresh % - Hyst %).

Example

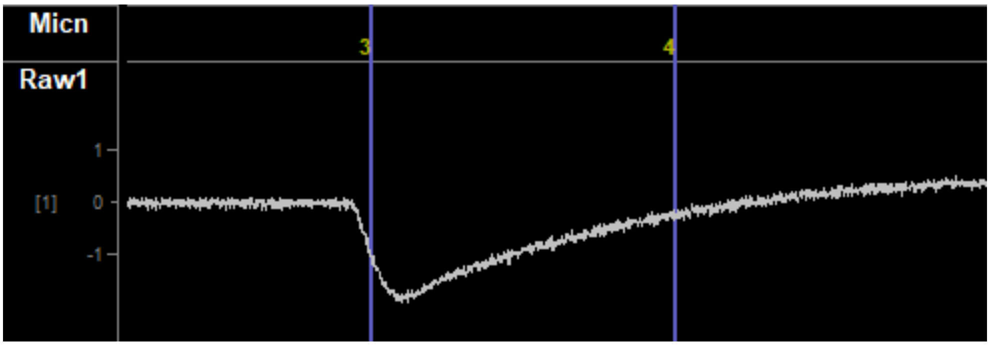


iM10 Raw Input Signal and Simple Threshold Crossing

Thresh is 20% and Hyst is 15%. The input logic is true when the signal crosses above 20% of the input range (20% of 5 V = 1 V) and false when the signal goes below 5% of the input range (0.25 V).

If threshold is negative, the lower threshold is (Thresh + Hyst).

Example



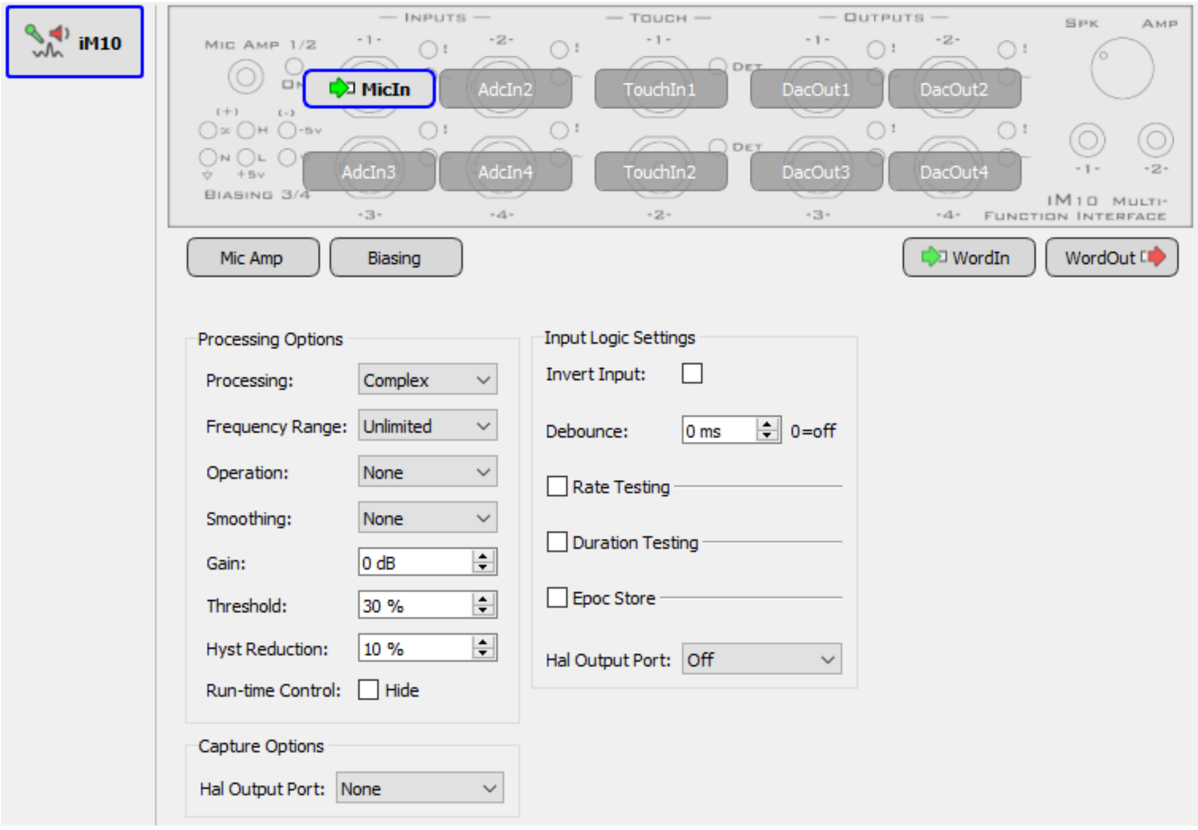
iM10 Raw Input Signal and Simple Threshold Crossing

Thresh is -20% and Hyst is 15%. The input logic is true when the signal crosses below -20% of the input range (-20% of 5 V = -1 V) and false when the signal goes above -5% of the input range (-0.25 V).

The rest of the processing is handled by the Logic Input Processor.

Complex

The iM10 processor runs at 400 kHz, independent of the RZ sampling rate. Use the Complex options to significantly increase fidelity in the frequency range you are interested in by reducing the input bandwidth. Also apply non-linear operations such as absolute value, square, flip the signal sign, or add smoothing.



iM10 Complex Input Options

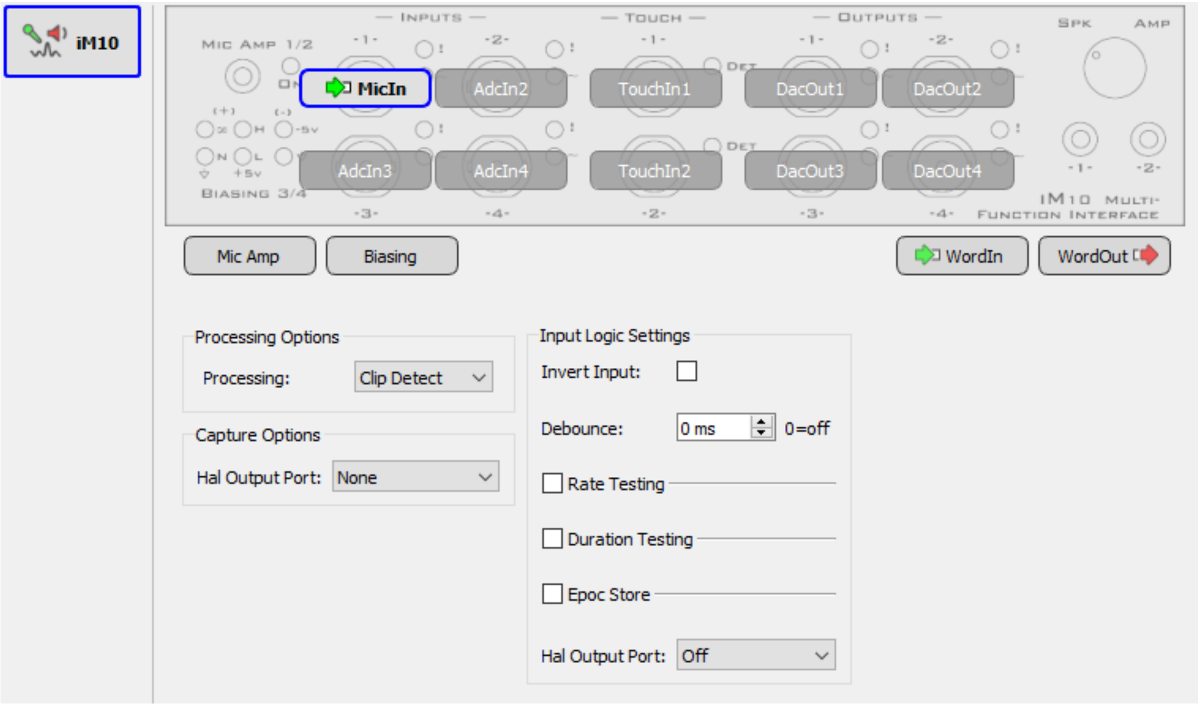
Option	Description
Frequency Range	Limit the frequency range of the input
Highpass Freq	Set highpass filter frequency (if `Frequency Range` is enabled)
Lowpass Freq	Set lowpass filter frequency (if `Frequency Range` is enabled)
Operation	Take the Absolute Value, Square the signal, or Flip the sign on the input
Smoothing	Set the time constant for exponential smoothing of the input signal

> **Note**
>
> All settings except for `Frequency Range` and `Operation` are adjustable in the run-time interface.

See Simple processing options for a description of the rest of the settings and thresholding examples.

The rest of the processing is handled by the Logic Input Processor.

Clip Detect



iM10 Clip Detect Input Options

Triggers when signal is within ~3% of the 5 V input range (greater than ~±4.8 V).

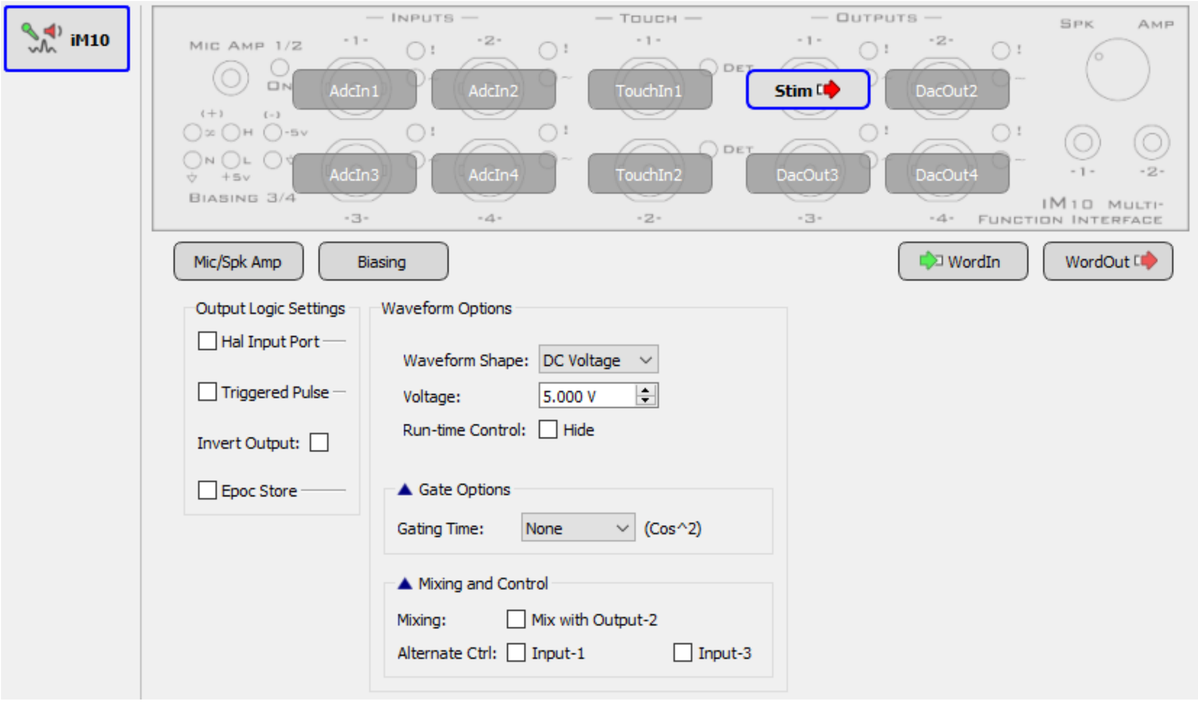
The rest of the processing is handled by the Logic Input Processor.

Touch Inputs

All touch sensor inputs are handled by the Logic Input Processor.

Analog Outputs

All iM10 outputs first go through a logical output processor that controls when the output is presented. See Logic Output Processor for more details.



iM10 Output Options

Waveform Shape	Description
User	Send an output signal from another gizmo's output link to the iM10
DC Voltage	Output a DC voltage when triggered
Tone	Generate up to 100 kHz tone **^**
White Noise	Generate a band-limited white noise
Pink Noise	Generate pink noise (1/f)
Square	Generate up to 100 kHz square wave (50% duty cycle, ±5 V) **^**
Clock	Generate up to 100 kHz clock signal (50% duty cycle, +5 V) **+**
PWM	Generate a pulse-width modulated waveform using 400 kHz clock (2.5 us minimum pulse width)

Option	Waveform Shape	Description
Gating	All	Add an optional cos2 gate to the waveform
Phase	Tone, White Noise, Pink Noise, Square, and Clock only	When `Frozen` is checked, the phase of the waveform resets to 0 when triggered
Freq Resolution	Tone, Square, and Clock only	Adjusting the frequency resolution changes the maximum frequency that the iM10 can generate **^**
Time Resolution	PWM only	Adjusting the time resolution changes the maximum period that the iM10 can generate for the PWM waveform
Attenuation	Tone, White Noise, Pink Noise, Square, Clock, and PWM only	iM10-generated waveforms use a 5 V amplitude. Apply up to 50 dB attenuation to the signal, adjustable at run-time.

^ For audio applications, set Freq Resolution to 10 Hz or 100 Hz only.

+ For precision control applications, use PWM instead

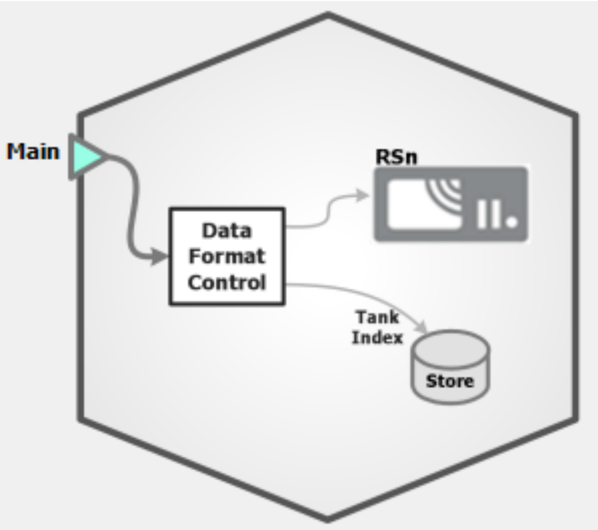
Outputs 1 and 2 are paired. Signals generated by Output 1 can be mixed with Output 2 and vice versa. Also, the choices of waveform shape for Output 2 depends on the selected waveform shape for Output 1. Likewise, Outputs 3 and 4 are also paired.

Output 1 & 3	Choices for Output 2 & 4
User	User, DC Voltage, White Noise, Pink Noise
DC Voltage	User, DC Voltage, White Noise, Pink Noise
Tone	DC Voltage, Tone, White Noise, Pink Noise, Square, Clock
White Noise	DC Voltage, Tone, White Noise, Pink Noise, Square, Clock
Pink Noise	DC Voltage, Tone, White Noise, Pink Noise, Square, Clock
Square	DC Voltage, Tone, White Noise, Pink Noise, Square, Clock
Clock	DC Voltage, Tone, White Noise, Pink Noise, Square, Clock
PWM	DC Voltage, White Noise, Pink Noise, PWM

RS4 Data Streamer

Important

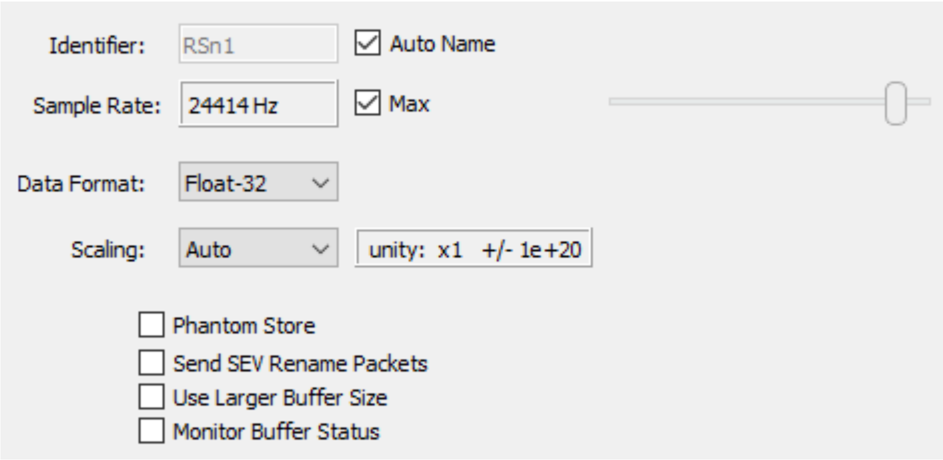
The RS4 is deprecated, consider the RS3 module as a replacement for high bandwidth data acquisition.



RS4 Block Diagram

The RS4 is a storage device with a fiber optic connection to an optical DSP on your RZ device. It is likely also connected to the same network as the computer running Synapse.

RS4 Options



RS4 Options

By default, the RS4 uses the RZ device sampling rate. You can set a rate manually by clearing the **Sample Rate Max** check box to enable the slider. You can also select the data format and any scaling. If streaming from an analog PZ5 or PZA amplifier, the Float-32 format is recommended due to the large input range and high fidelity. If using a ZD digital headstage or PZ2 amplifier, Integer-16 format with Scaling set to 1e6 is preferred.

Phantom Store generates header information in the data tank. This is useful if you are reading the RS4 data with TDT applications through the tank server. However, this consumes extra hard disk space on your Synapse computer, so if you are going to read the files directly from the RS4 for your own custom analysis then leave this unchecked.

Send SEV Rename Packets sends a UDP packet containing the tank and block name over the local network to the RS4, so that the RS4 can rename the data files correctly.

When **Phantom Store** is selected, **Send SEV Rename Packets** is selected for you.

> **Important**
>
> If you aren't using TDT applications to read the data saved on the RS4, you don't need to select **Phantom Store** but you ***absolutely*** should select **Send SEV Rename Packets** so that the RS4 data is organized properly.

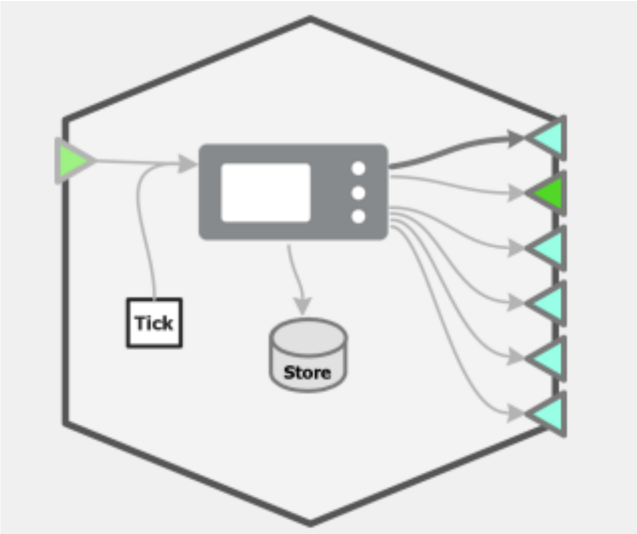
Use Large Buffer Size allocates more DSP memory to buffer the data before it is transferred to the RS4. This improves reliable when streaming very high bandwidth, high channel count data to multiple RS4 ports. The RS4 should be updated to v1.19 firmware or above to most effectively use this setting.

Monitor Buffer Status adds a runtime interface that warns you of data errors for high bandwidth streaming applications (64+ channels at fast storage rates). It is not accurate for lower bandwidth streaming (32 channels and below or slow storage rates).

> **Note**
>
> See the RS4 section of the System 3 Manual for more information about Data Streamer operations.

RV2 Video Tracker

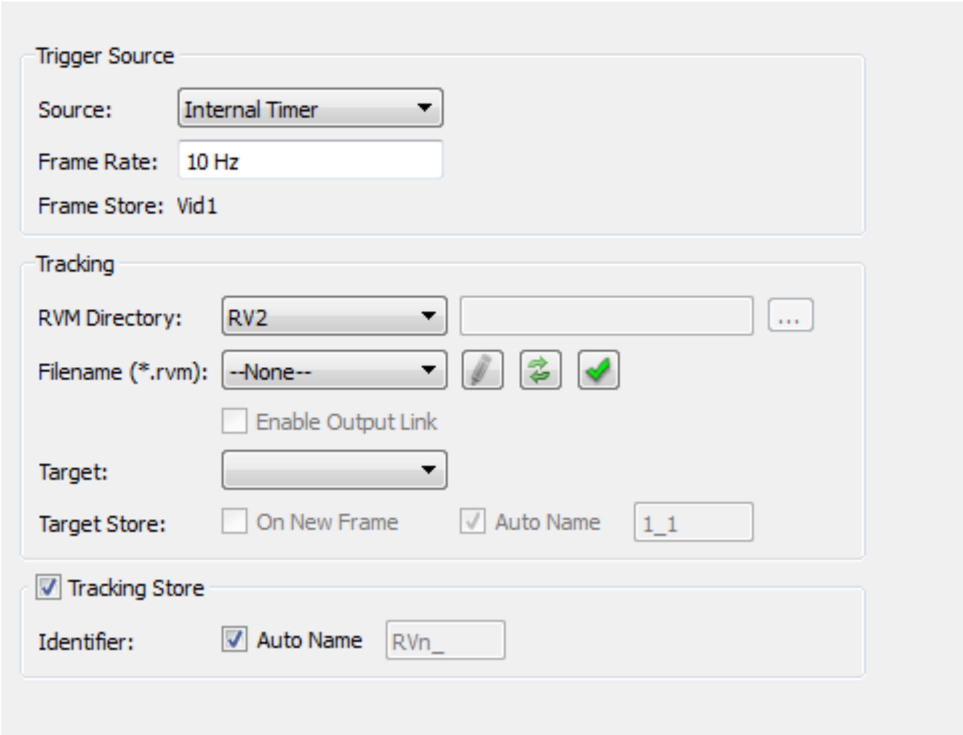
> **Important**
>
> The RV2 is deprecated, consider the iV2 module as a replacement for precisely timed frame acquisition and real-time subject tracking.



RV2 Block Diagram

The RV2 receives triggers from the RZ, processes video frames, and returns tracking information to the RZ for storage and/or further processing. The RV2 must be on the same local network as the PC running Synapse. Its IP address must be set in the Rig Editor.

RV2 Options



RV2 Options

The trigger source can be either an internal clock or an input from another gizmo. The frame number is timestamped and stored to disk. Make sure the frame rate is less than the free run frame rate which is displayed on the RV2 Live tab. Otherwise you will see frames dropped and missing from your video file.

RVM files define the tracking algorithm and the tracked data that the RV2 returns. They are created in RVMap software, which installs with TDT Drivers, and are stored in the **configs** directory on the RV2. The Filename list contains the RVMs found on your RV2 device. Choose the appropriate one for your experiment.

The buttons next to the list perform the following actions:

Button	Action
	Open the selected file in RVMap for editing
	Refresh the configurations list
	Send the selected map file to the RV2 to preview the tracking algorithm on the RV2 screen

> **Note**
>
> Whatever RVM file is selected is also sent to the RV2 when you exit designtime. The RVM Directory allows you to select a local directory of RMV files for offline experiment design when the RV2 is not available.

Each RVM file defines a number of tracked targets. For each **Fixed** or **Relative** target defined in the RVM file, positional data consisting of X, Y, and region values is returned to the RZ on each frame. **Reference** targets also contain heading information.

When an RVM file is selected from the drop-down list, the target information is parsed and the total number of data points is displayed next to the **Channel** label and the **Target Selector** drop-down is updated with the list of available tracked targets.

When **Enable Output Link** is selected, the target information is available for real-time processing by other gizmos. If the RVM file contains a target called 'T1', and 'T1' is selected in the **Target Selector**, then data sources 'T1X', 'T1Y', 'T1R' can be used by other gizmos for real-time processing. For example, if you want to trigger an event when the subject is in a particular region, you can feed the 'T1R' output to the State Maker gizmo and use it to choose an outcome based on subject position, all in real-time.

All target positional information is output on the **AllTracking** output link, so if you need to extract more than one target for real-time processing you can connect a Signal Selector gizmo to the **AllTracking** link and pick off a signal channel for additional real-time processing.

The **Tracking Store** option saves all target positional data to disk as a Strobe Store when the frame is received. The **Selected Target** option creates an Epoch Store with information from the target chosen in the **Target Selector** drop- down.

> **Note**
>
> See the RV2 section of the System 3 Manual for more information about Video Tracker operations and RVMap software.

Gizmo Reference

Gizmo Reference

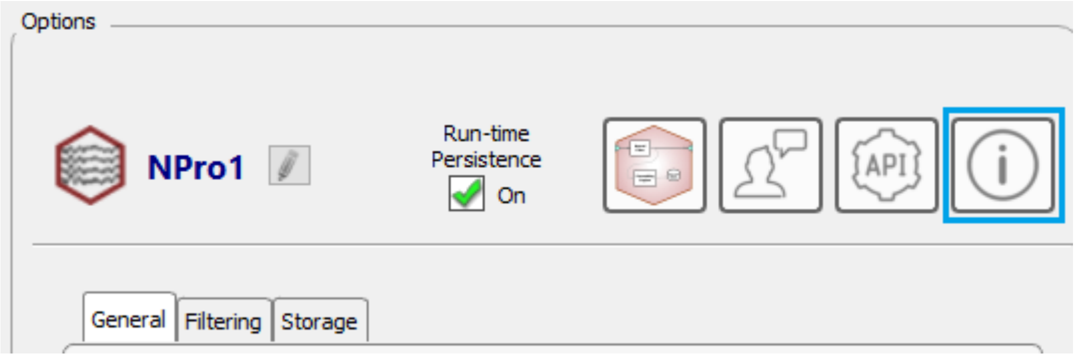
Getting Help

Each gizmo reference in this manual includes four sections:

1. **Common Use Cases** A brief description, common use cases, and common inputs and outputs. For a list of all gizmos in this manual organized by category, see Gizmo Categories.
2. **Gizmo Help Slides** These provide a bit more in depth information on each each gizmo. These are also available directly in Synapse when browsing gizmos. See Gizmo Slides Window below.
3. **Examples** Any available examples that include this gizmo are listed here.
4. **Reference** The rest of each gizmo section is specific details for working with each gizmo.

From Within Synapse

You can access this manual directly within Synapse by clicking the information icon shown in the Options icon bar of any gizmo. This opens the web version of the Synapse Manual. If there is currently no internet access, it opens the local PDF and jumps you to the page for the gizmo you are currently looking at.

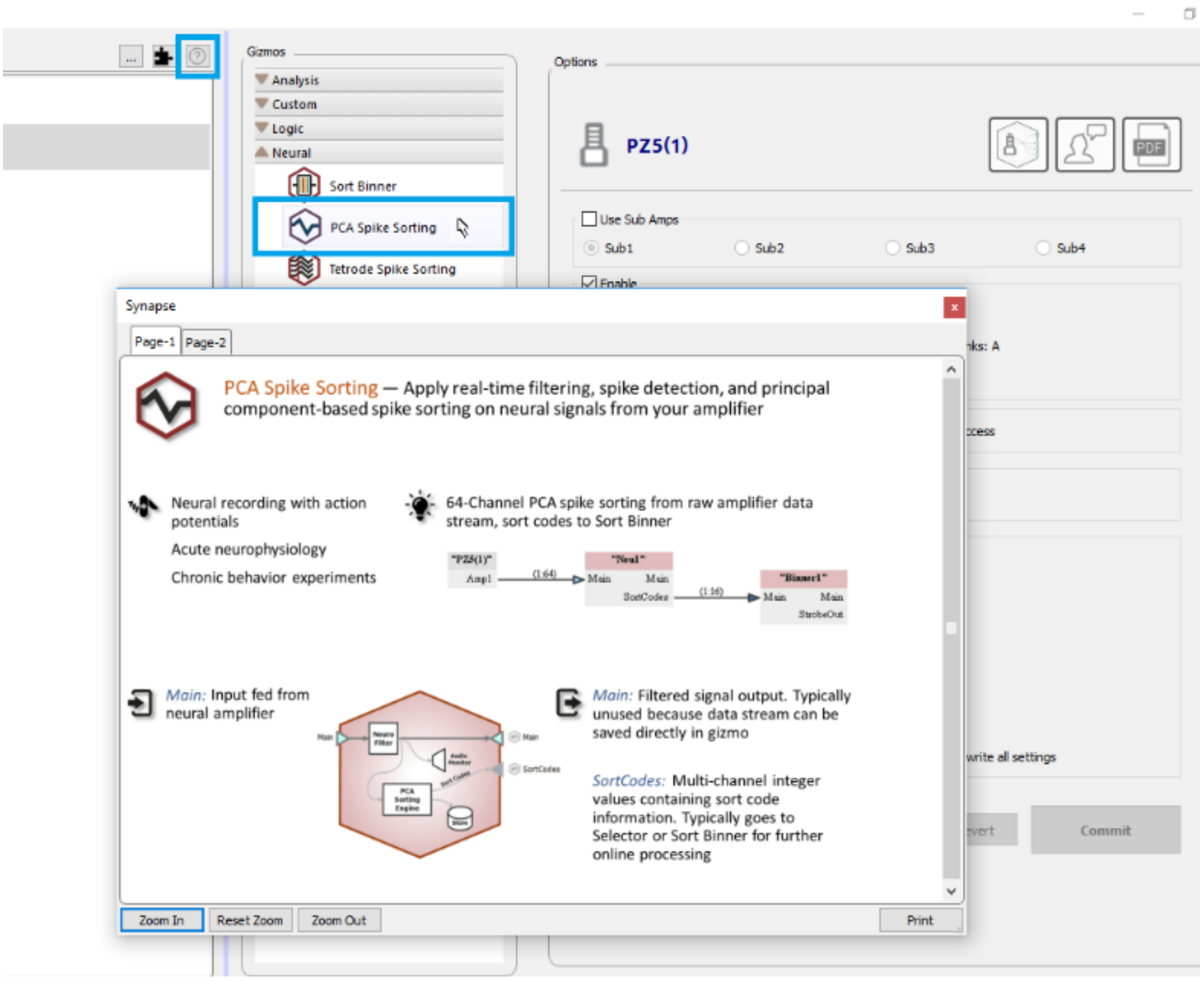


Gizmo Slides Window

The Gizmo Slides Window provides a top-level view of the currently selected gizmo. The slides window is available by clicking the '?' button above the Processing Tree.

Each gizmo has a set of slides that show common use cases, example connection diagrams, common inputs and outputs, an overview of the user interface, common API parameters, and more.

When you select a new gizmo, the slides for that gizmo are shown. This is helpful when you are getting to know the user interface and what all the gizmos do.



PCA Spike Sorting Gizmo Slide Example

Here is a legend for the common icons found in the gizmo slides:

Common use cases

Example connection diagram

Gizmo inputs

Gizmo outputs

User interface description

Configuration options

API parameters

Data stores

Gizmo Categories

Analysis Gizmos

Perform powerful real-time analysis on incoming signals.

Icon	Name	About this Gizmo	When to Use	Common Inputs & Outputs
	Signal Accumulator	Collect a sum of an incoming signal over a user-defined window. Optionally compute the average as well. Can also perform thresholding of accumulated signal for further processing	Use this gizmo to calculate the total power of a signal over a specified time span, or to compute average signal power over many trials	Inputs any single or multi-channel data stream. Outputs the sum of the accumulated signal and optionally the average.

Logic Gizmos

Combine logical signals originating from external hardware or other gizmos into meaningful logic states.

Icon	Name	About this Gizmo	When to Use	Common Inputs & Outputs
	State Maker	State Maker is an interface for performing logical tests on single-channel inputs and combining results into output states for storage and controlling/ triggering other gizmos for further signal processing	Use this gizmo when receiving bit codes from external devices, and to make decisions/ process gizmo output values. Used often to trigger store events or strobing other gizmos	Digital I/O inputs from RZ gizmo, or inputs from other gizmos, such as integer values from a Selector reading sort codes. Output variable duration logic triggers based on a combination of keys and marks
	Timer	Measures time between or duration of logical events from primary and secondary inputs	Use this gizmo to calculate event frequency or time logical events. Can be used to measure response time to stimuli, calculate heart rate, and time other physiologic intervals	Inputs any logic signals. Outputs smoothed or instantaneous measurements of period or frequency.
	Pulse Generator	Creates user-defined pulse trains based on milliseconds or Hz. Control duty cycle, period, number of pulses, and trigger pulse trains internally or from other gizmos	Use this gizmo for directly controlling optogenetic stimulation or driving the timing of other connected gizmos or devices	Inputs logic strobe signal for other gizmos or digital I/O. Output TTL or floating-point pulses to be routed to other gizmos or I/O
	User Input	Create dynamic stores and logic outputs based on inputs from digital I/O bits or a software button	Use this gizmo to store I/O inputs with values defined by user or another gizmo and create fixed-duration, toggled, or edge logic outputs based	Input from digital I/O bits or software button. Output logic as an edge, fixed-duration TTL pulse, or toggled output. Output can be a counter, a user-set value, or gizmo input
	Pulse Train Generator	Create simple or complex user-defined pulse train waveforms based on a number of configuration options that include stacked or parallel trains. Control whether trains are strobed or triggered, and define waveform characteristics with a parameter table.	Use this gizmo for directly controlling optogenetic stimulation or other connected gizmos or devices. Pulse Train is especially useful when you want to create complex pulse trains where one signal is gating another, or if you want to inform waveform parameters dynamically from other gizmos.	Input from logic strobe signals, parameter values, or scalar inputs. Output a TTL or floating point train signal to driver connected gizmos or devices.

Neural Gizmos

Combine logical signals originating from external hardware or other gizmos into meaningful logic states.

Icon	Name	About this Gizmo	When to Use	Common Inputs & Outputs
	PCA Spike Sorting	Real-time filtering, spike detection, and principal component-based spike sorting with selectable algorithms	This is the most common method for online spike sorting. Cluster units in PCA space and identify spikes automatically or manually cut	Input any multi-channel neural stream (raw amplifier stream). Output integer sort codes.
	Tetrode Spike Sorting	Real-time filtering, cross-channel tetrode spike detection and classification in a fully customizable 2D feature projection	Use this gizmo for sorting spikes using tetrodes. Commonly used for cell isolation, tetrode sorting provides high spatial localization of nearby units	Input any multi-channel neural stream (raw amplifier stream), often from a mapper to organize tetrode channels. Output integer sort codes.
	Box Spike Sorting	Real-time filtering, spike detection, and discrimination of neural signals using time-voltage windows	Use this gizmo to sort neuronal spikes on individual channels using time-voltage discrimination windows	Input any multi-channel neural stream (raw amplifier stream). Output integer sort codes
	Sort Binner	Compress sort code output from spike sorting gizmos for fast viewing and further processing. Optionally output to RZ UDP interface for external processing	Use this gizmo to count the number of sort codes that occur on specific channels within a user-specified time window	Input from sort code outputs of spike sorting gizmos. Output 32-bit integer words that are a count of sort codes per channel.
	Neural Stream Processor	Easily visualize, filter and store real-time multichannel neurophysiology signals. Includes built in, optimized settings for the most common biologic signal types	Use this gizmo for easy filtering and storage of common signal types: LFP, EEG, EMG, Single-Unit, EKG	Input any neural stream (typically the raw signal). Outputs filtered signal, and also saves the filtered signal by default.
	Neural Signal Referencer	Digitally subtract common signals from multi-channel stream. Single or multi-channel referencing on all channels or independent sub-groups of channels	Use this gizmo to eliminate common mode noise across channels or to perform digital re-referencing. Multi-channel referencing won't create artificial waveforms on your signal	Input any multi-channel neural stream (typically pre-filtered). Output multi-channel re-referenced signal, and save reference signal itself.

Specialized Gizmos

Specialized gizmos encapsulate a specific application all in one gizmo.

Icon	Name	About this Gizmo	When to Use	Common Inputs & Outputs
	Fiber Photometry (RZ10x)	Real-time control and acquisition of demodulated locked-in amplification signal from any combination of up to 3 light drivers and 2 photosensors on a Lux bank. Can also monitor light power from PM1 from any Lux bank	This is the primary gizmo used in fiber photometry setups using RZ10x processors. Record up to 6 demodulated signals with raw photosensor output and dFF calculations, too.	Input from the RZ #enable line or other logic strobe signals. Output calculated Driver x Sensor signals or dFF signals
	Fiber Photometry (Legacy)	Real-time control and acquisition of demodulated locked-in amplification signal from any combination of up to 4 light drivers and 2 photosensors	This is the primary gizmo used in fiber photometry setups using processors other than an RZ10x. Record up to 8 demodulated signals with raw photosensor output too	Input from the RZ #Enable line or other logic strobe signals as master LED control. Output demodulated Driver x Sensor signals
	MRI Recording Processor	Suppress MRI recording artifacts using controllable signal gate. Titrate gating tightly around artifact to clean up online signals	Use this gizmo to eliminate gradient switching artifact in an MRI recording environment. Can automatically detect artifacts or be triggered using timing signal from the magnet	Input any multi-channel neural stream. Output multi-channel filtered and artifact-free single unit and LFP signals
	Python Coding Gizmo	The Python Coding Gizmo ("Pynapse") tightly integrates Python coding into your Synapse experiment	Use this to control your experiment flow, build and deliver stimuli, run complex behavioral paradigms, do custom analysis and visualization	Input up to 8 logic or number signals for logic testing, output logic or custom waveforms

Routing Gizmos

Work with single or multi-channel signals in the Synapse framework. Control signal flow.

Icon	Name	About this Gizmo	When to Use	Common Inputs & Outputs
	Mapper	Create user-defined channel maps to reroute electrode sites to different amplifier channels	Use this when you want to create ordered spatial maps with unordered electrode sites	Input any multi-channel stream, typically right from the amplifier. Output a reordered multi-channel stream
	Selector	Pick off individual channels from a multi-channel stream, or isolate specific channel and sort code combinations from Sort Binner	Use this gizmo for routing individual channels for monitoring or further processing, or for reading Sort Binner outputs of single channel + sort code information	Input any multi-channel signal or Sort Binner. Output the isolated channel for further processing or the Sort Binner count of sort code occurrences on the specified channel
	Merger	Combine up to eight single or multi-channel streams into a single multi-channel stream	Use this gizmo to send separate data into a single multi-channel stream for processing in other gizmos or storage	Two or more single or multi-channel data streams. Must be of the same single or multi-channel type. Output the merged data streams
	Injector	Insert a single channel input into a multi-channel data stream at specific user-specified channels	Choose a channel for electrical stimulation. Can also be used to route audio signal to a speaker array (channel in DAC Montage).	Input a single channel input such as eStim or aStim. Output into a multi-channel signal with channel routing information
	Delay	Adds a fixed or dynamic delay to any input signal	This gizmo is useful for triggering optogenetic, auditory, or other stimuli a programmed time after an event of interest occurs	Input from any signal. Output the same signal at a specified time later
	Parameter Manifold	Control multiple stimulation gizmo parameters simultaneously	Use this gizmo when needing to share parameters between multiple stimulation gizmos, such as duration or pulse count. Often used in conjunction with Parameter Sequencer	Strobe signal input from other gizmos. Output shared parameter values to multiple gizmos

Signal Conditioning Gizmos

Perform signal conditioning and processing on incoming data.

Icon	Name	About this Gizmo	When to Use	Common Inputs & Outputs
	Unary Processor	Implement series of mathematical operations to incoming signals	Use this gizmo to perform interesting signal processing on incoming data, such as power in band, RMS, or scaling. Can also perform complex thresholding or type conversion on signals	Input any single or multi-channel data stream. Outputs the processed data stream or a converted signal type
	General Purpose Filter	Create filters with user-defined parameters that include high/ low pass corners up to 8^{th} order and notches with varying cut depths and bandwidths	Use this gizmo to design a filter with higher orders or more notches than the Neural Stream Processor can provide	Input any single or multi-channel data stream. Outputs the filtered data stream
	Artifact Blocker	Suppress artifacts associated with triggered events. Includes gate timing parameters for control of gate shape	Use this gizmo to remove large artifacts during events like electrical stimulation or motion artifact	Any single or multi-channel data stream. Outputs the same signal, but with the data removed around the artifact event

Stimulation Gizmos

Stimulation gizmos generate precisely sequenced audio, electrical, or optical stimulation.

Icon	Name	About this Gizmo	When to Use	Common Inputs & Outputs
	Parameter Sequencer	Control stimulus parameters with complex timing and presentation sequences (rolling, repeated, random, manual)	High-level parameter control and stimulus presentation	Root or strobe input from another gizmo. Outputs Parameter value and strobing logic
	Audio Stimulation	Generate fully customizable tone, noise, and other audio stimulation types	Use this gizmo for audio neurophysiology, stimulus-response protocols, hearing screening protocols, and psychoacoustics	Strobe inputs; parameter inputs from Parameter Sequencer. Output the audio signal and a stim sync logic signal
	Electrical Stim Driver	Create up to four stimulation voices for single-ended or bipolar stimulations outputs on a target device, such as an IZ2 or IZV. Create monophasic or biphasic waveforms with charge balancing options.	Use this gizmo for design of interesting electrical stimulation waveforms	Strobe inputs; parameter inputs from Parameter Sequencer. Output voices to target stimulation devices
	File Stimulation	Play custom waveforms from a list of files on disk, which includes WAV files and MAT files	Use this for speech studies, psychoacoustics, or for custom audio or electrical stimulus presentations	Strobe inputs; parameter inputs from Parameter Sequencer. Output stimuli and a stim sync logic signal
	Ultrasonic File Stimulation	A streamlined version of File Stimulation for creating custom stimuli at ultrasonic frequencies	This gizmo is useful for audio neurophysiology and stimulus response protocols for animals that can hear in the ultrasonic frequency range	Strobe inputs; parameter inputs from Parameter Sequencer
	Ultrasonic Stimulation	A streamlined version of Audio Stimulation for creating stimuli at ultrasonic frequencies	This gizmo is useful for audio neurophysiology and stimulus response protocols for animals that can hear in the ultrasonic frequency range	Strobe inputs; parameter inputs from Parameter Sequencer

Storage Gizmos

Precisely timestamp and store any type of real-time data to disk.

Icon	Name	About this Gizmo	When to Use	Common Inputs & Outputs
	Stream Data Storage	A general-purpose gizmo used to store single or multi-channel data streams. Includes data formatting and scaling options	Use this gizmo to store raw data directly from your amplifier. Use on the output of other gizmos that do not have storage options, like Unary Processor	Input any single or multi-channel data stream. No outputs
	Strobed Data Storage	Store single values or short segments of data (including pre-trigger data). Includes heat maps and bar plots	Use this gizmo to store streaming data asynchronously or store values/ segments of data around events of interest	Input any single or multi-channel data stream and a strobe input. No outputs
	Epoch Event Storage	Timestamp and store single or multi-channel data when triggered	Use this gizmo to capture behavioral inputs or stimulus parameters to filter and align neurophysiological data	Input any data input and a strobe input. No outputs

Visualization Gizmos

View incoming signals in dynamic ways and perform interesting processing on them

Icon	Name	About this Gizmo	When to Use	Common Inputs & Outputs
	Oscilloscope	Has all the functionality of a hardware oscilloscope and more. View up to four channels at user-defined ranges and domains, and perform complex signal testing for creating trigger outputs	Use this gizmo to visualize signals on a more refined time scale, or to perform thresholding or hysteresis tests for complex triggering paradigms like phase-locked stimulation off LFPs	Input any single-channel signal. Outputs logical triggers and delayed signal

Custom Gizmos

Create your own custom real-time signal processing function

Icon	Name	About this Gizmo	When to Use	Common Inputs & Outputs
	User Gizmo	Use the RPvdsEx coding environment to make a custom gizmo with any component in the RPvdsEx library	When you can't find a standard gizmo to do what you want	Any single or multi-channel data stream

Using Parameters

A bounded parameter is a named value that controls a parameter in the underlying real-time processing of the gizmo that contains it. The parameter's value can be modified at runtime by the user or by another gizmo. The user sets the allowed minimum and maximum values of each bounded parameter at designtime. These values are enforced whenever the parameter value is modified.

Gizmos that support bounded parameters share a common runtime interface which gives you manual, semi-automated or fully-automated control of the parameters at runtime.

The stimulation gizmos share a common set of bounded parameters with consistent names to define and organize information about the stimulus parameters so that you can easily switch between them. All of the stimulation gizmos and many of the routing gizmos use parameter tables.

Video Demonstrations

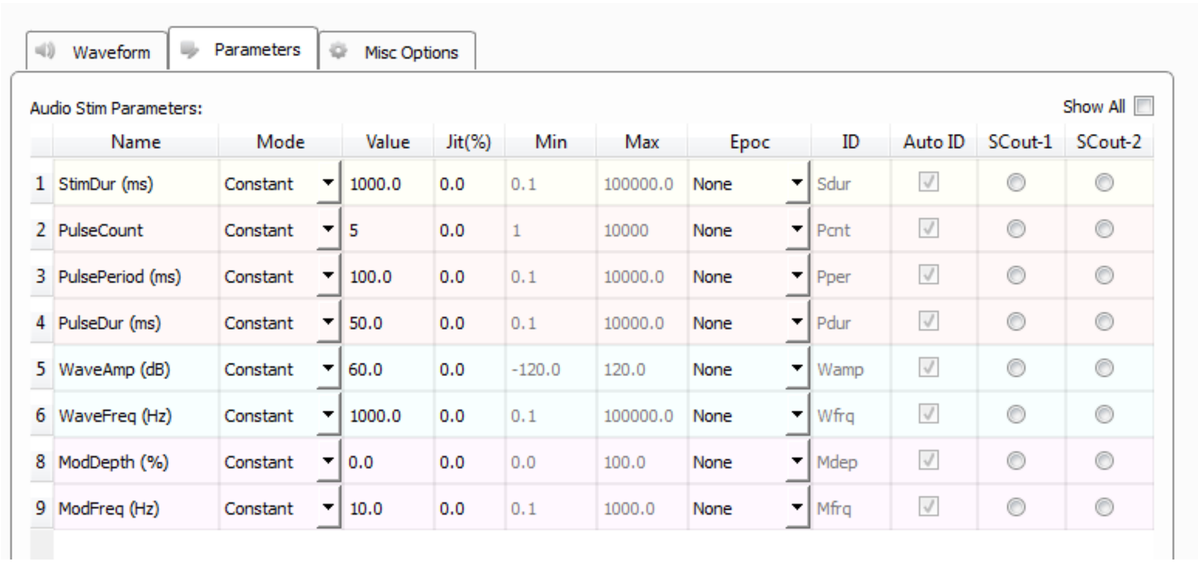
Follow along with these videos for a full experiment walkthrough using signals and parameters.

Signals & Parameters Part 1

Signals & Parameters Part 2

Parameters Table

Gizmos that support bounded parameters have a Parameters tab in their designtime interface that contains the parameters table. The parameters table contains all of the possible bounded parameters for that gizmo. The Audio Stimulation gizmo parameter table is shown in the example below.



	Name	Mode	Value	Jit(%)	Min	Max	Epoc	ID	Auto ID	SCout-1	SCout-2
1	StimDur (ms)	Constant	1000.0	0.0	0.1	100000.0	None	Sdur			
2	PulseCount	Constant	5	0.0	1	10000	None	Pcnt			
3	PulsePeriod (ms)	Constant	100.0	0.0	0.1	10000.0	None	Pper			
4	PulseDur (ms)	Constant	50.0	0.0	0.1	10000.0	None	Pdur			
5	WaveAmp (dB)	Constant	60.0	0.0	-120.0	120.0	None	Wamp			
6	WaveFreq (Hz)	Constant	1000.0	0.0	0.1	100000.0	None	Wfrq			
8	ModDepth (%)	Constant	0.0	0.0	0.0	100.0	None	Mdep			
9	ModFreq (Hz)	Constant	10.0	0.0	0.1	1000.0	None	Mfrq			

Parameters Tab

This table allows the user to set the parameter source and bounds at designtime, among other things that are discussed below.

Rows are shown or hidden depending on the stimulus type and in response to selections made during configuration, with only relevant parameters shown. Likewise, the columns contain values to further define the parameter and are enabled or disabled (gray) by choices you make during the design process.

Value and Min/Max

Value sets the default value for the parameter when you switch to runtime mode.

All parameters are bounded by their **Min** and **Max** values. Whenever possible, narrow the bounds to the most reasonable values for the parameter.

Min and **Max** set the bounds on any runtime slider widgets and inform any upstream gizmos, such as Parameter Sequencers, about the required values.

Right-click in the Value cell to open a pop-up dialog for easier value entry.

Epoc and ID

In the Epoc column, you can choose to save the individual parameter value on a strobe event or on value change. The options differ depending on the type of gizmo.

Synapse automatically generates a store name. TDT recommends using Auto ID to ensure no store names are duplicated. A "/" is appended to the name to indicate when the full epoc is stored. To make your own store names, clear the **Auto ID** check box.

Mode

In the Mode column, you can choose to make individual parameters constant, dynamically controlled by a runtime **Widget** (slider), dynamically controlled by a parameter input (**Param In**) from a Parameter Sequencer gizmo or Parameter Manifold gizmo, or dynamically controlled by one of two possible single channel gizmo inputs (**Scalar In-1**, **Scalar In-2**).

Constant

In constant mode, **Value** defines the value of the parameter. The value can be seen in the runtime interface, but cannot be changed.

Jit%, or percentage jitter, acts as a randomizer for each presentation. If non-zero, then **Min** and **Max** provide the bounds.

Widget

Widget Run-time Interface

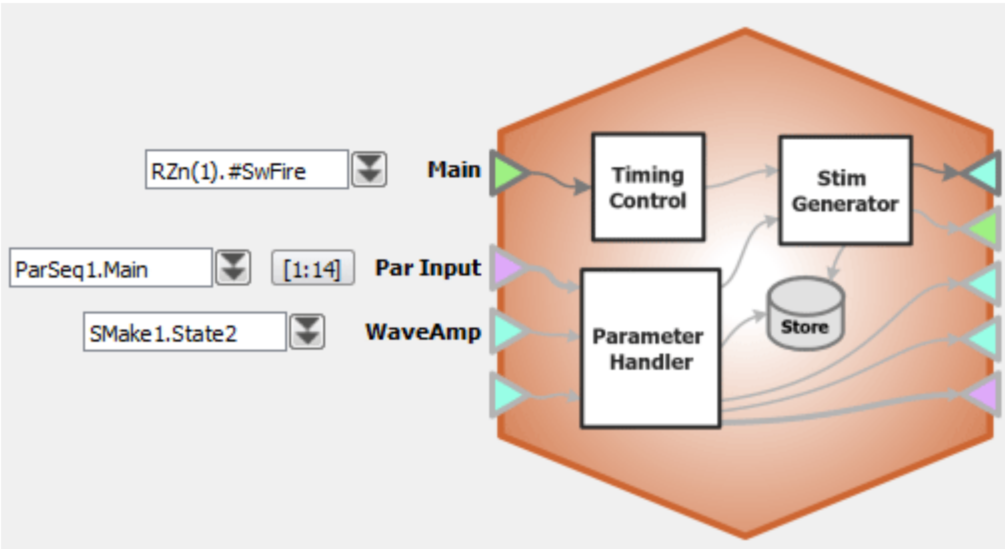
In Widget mode, **Value**, **Jit%**, **Min**, and **Max** define the reasonable limits for the parameter and set the initial value.

At runtime a interface is added as a tabbed window that includes a value box and slider for the parameter(s) set to **Widget** in the table. A manual **Strobe** button presents a single stimulus. A mute button zeros the signal when checked.

Scalar Inputs

Scalar Inputs 1 and 2 provide a gizmo input to control a parameter directly from any other floating point signal within the bounds defined for the parameter

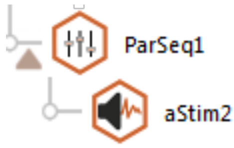
> **Tip**
>
> You must set the parameter up in the table and Commit the change, then update the source for the new input in the block diagram.



Block Diagram with Several Stimulus Parameter Inputs

Param In

In **Param In** mode, the parameter value is read in from another gizmo. **Jit%** (Jitter), **Min**, and **Max** primarily define the reasonable limits for the variable. **Param In** is intended specifically for use with the Parameter Sequencer gizmo or Parameter Manifold gizmo. You will need to configure the stimulus, and choose the **Param In** mode before adding the Parameter Sequencer or Parameter Manifold gizmo to the Processing Tree. Once attached to the parent gizmo, the parameters from the stimulus will be automatically added to the parent's parameter table.

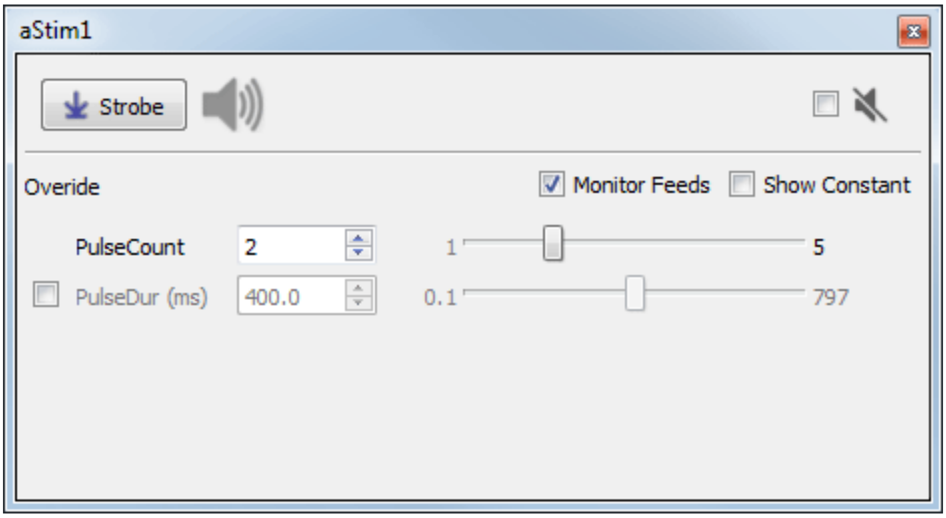


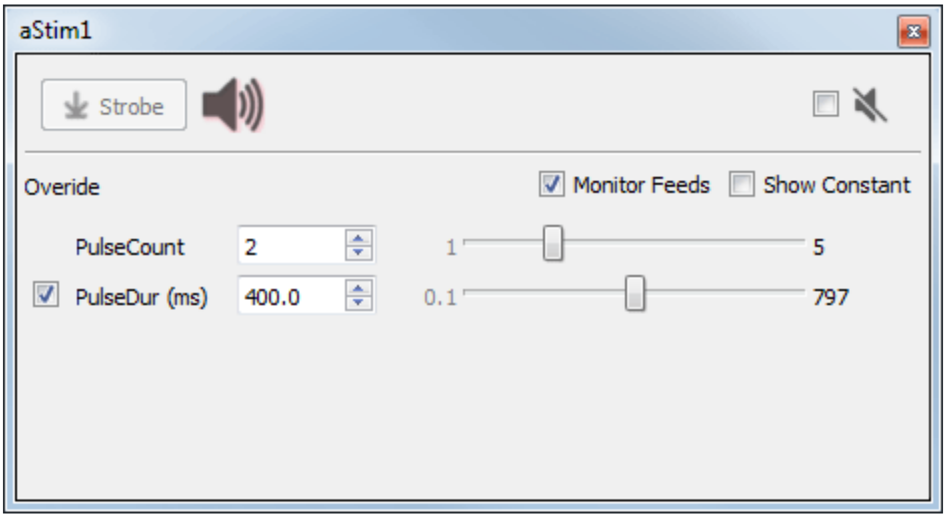
Parameter Sequencer and Audio Stimulation Gizmos

In the example below, Pulse Count is controlled by a widget and PulseDur is controlled by in a parameter input. The **Monitor Feeds** check box shows the parameter controlled input.

Notice, on the left, only the widget controlled parameter is editable. On the right, The check box in the **Override** column next to the PulseDur (the **Param In** controlled) parameter has been selected and is now editable.

This allows you to override the Parameter Sequencer inputs at any time.





Stimulation Runtime Tab Showing "Monitor Feeds" Enabled

Scalar Outputs

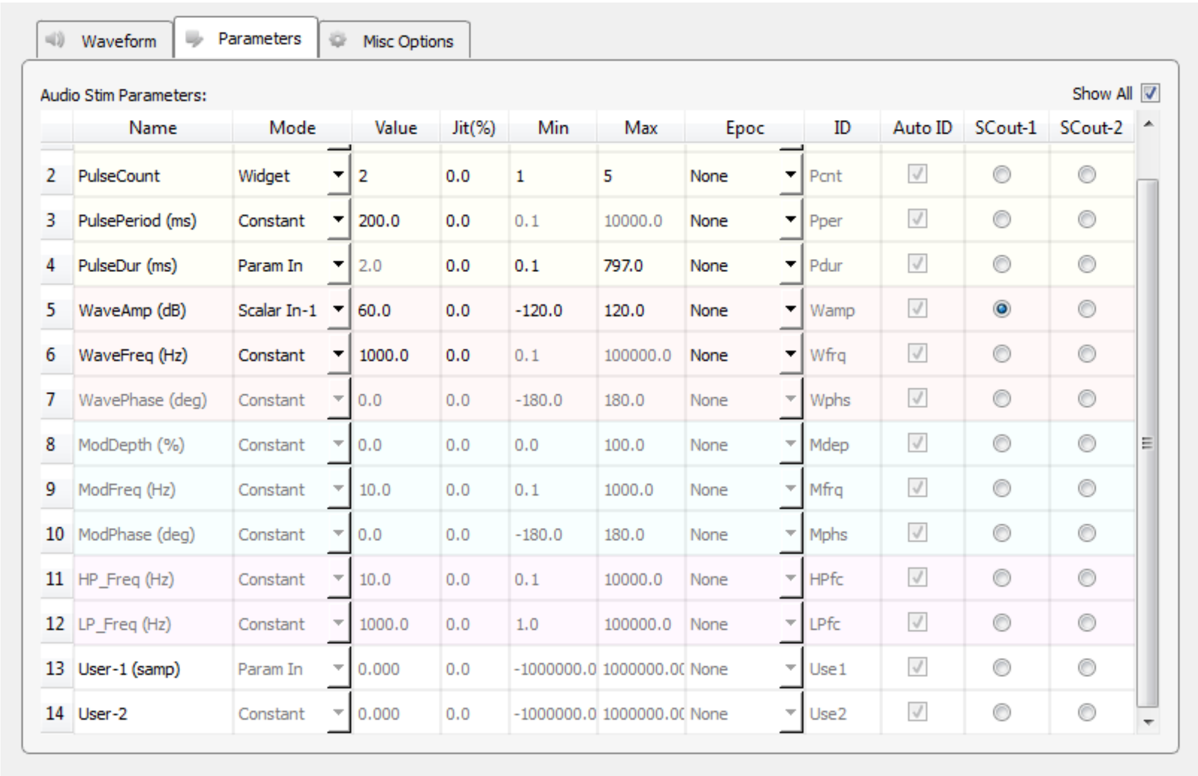
Use radio buttons in the **SCout-1** or **SCout-2** columns to select a parameter to output. The output can then feed an input on another gizmo. You must commit the change before the new output line will be enabled and labeled in the block diagram.

User Parameters

User-1 and **User-2** are parameters meant for your custom needs. These parameters can be virtually anything you need them to be. For example, they can be useful for defining a stimulus presentation channel controlled by the Sequencer.

To locate these parameters:

1. On the Parameters tab, select the **Show all** check box and scroll to the end of the list.

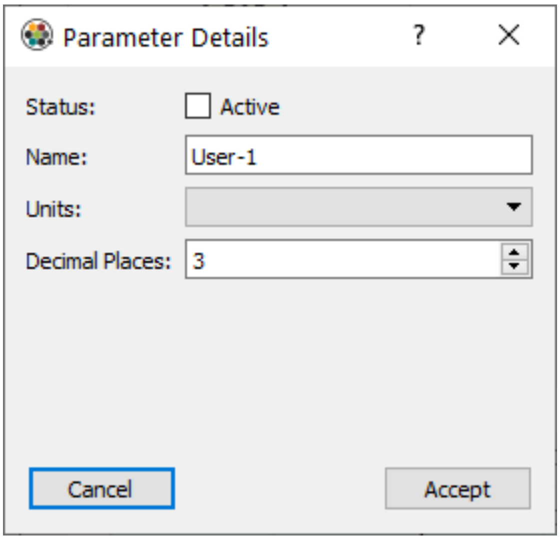


	Name	Mode	Value	Jit(%)	Min	Max	Epoc	ID	Auto ID	SCout-1	SCout-2
2	PulseCount	Widget	2	0.0	1	5	None	Pcnt			
3	PulsePeriod (ms)	Constant	200.0	0.0	0.1	10000.0	None	Pper			
4	PulseDur (ms)	Param In	2.0	0.0	0.1	797.0	None	Pdur			
5	WaveAmp (dB)	Scalar In-1	60.0	0.0	-120.0	120.0	None	Wamp			
6	WaveFreq (Hz)	Constant	1000.0	0.0	0.1	100000.0	None	Wfrq			
7	WavePhase (deg)	Constant	0.0	0.0	-180.0	180.0	None	Wphs			
8	ModDepth (%)	Constant	0.0	0.0	0.0	100.0	None	Mdep			
9	ModFreq (Hz)	Constant	10.0	0.0	0.1	1000.0	None	Mfrq			
10	ModPhase (deg)	Constant	0.0	0.0	-180.0	180.0	None	Mphs			
11	HP_Freq (Hz)	Constant	10.0	0.0	0.1	10000.0	None	HPfc			
12	LP_Freq (Hz)	Constant	1000.0	0.0	1.0	100000.0	None	LPfc			
13	User-1 (samp)	Param In	0.000	0.0	-1000000.0	1000000.0(	None	Use1			
14	User-2	Constant	0.000	0.0	-1000000.0	1000000.0(	None	Use2			

Parameter Table with Show All Selected

2. After you locate the User parameters, double-click the name cell to open the Parameter Details dialog box.

> **Important**
>
> You must select the **Active** check box to enable it.



Parameter Details Dialog Box

In the dialog, you can define some of the parameter's basic properties. Once the properties are accepted and the parameter is active, you can configure it much like you would any other parameter.

The Parameter Sequencer Gizmo

Stimulus gizmos uses uniform parameter tables to configure the stimulus parameters. Because the tables are structured using consistent parameter structure and naming, you can use a Parameter Sequencer gizmo or Parameter Manifold gizmo to feed values to one or more parameter tables in a systematic way.

Creating User Gizmos

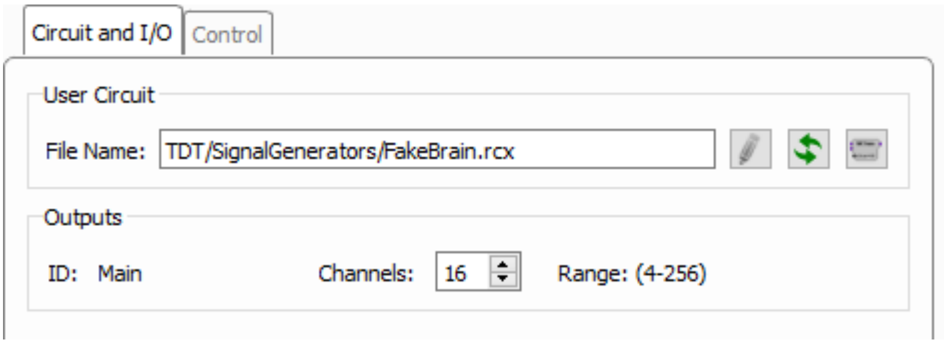
User gizmos are a class of gizmos which can bring customized processing tasks and user interfaces into Synapse. Any desired processing task that is not already defined in a provided gizmo can be created using user gizmos and then linked into the processing tree, just like any other gizmo.

User gizmo functionality is defined by 'circuits' that are designed in RPvdsEx software. The circuit defines what kind of inputs and outputs the user gizmo accepts and what type of user interface controls will display at designtime and/or runtime to dynamically modify parameters within that gizmo processing task.

Getting Started

Intro

User gizmos are *.rcx files created in RPvdsEx software. To add a user gizmo, add the **New User Gizmo** in the Processing Tree. The user gizmo interface has two tabs, Circuit and I/O and Control.



Circuit and I/O Tab

On the Circuit and I/O tab, you can use the first button to the right of the File Name field to browse to an RCX file. Once selected, Synapse parses that file and updates the user interface with any available options, such as user selectable channel counts. If there are any designtime controls specified in the RCX file, they will appear on the Control tab.

The other two buttons on the right allow you to edit the currently selected RCX file in RPvdsEx, and reload the selected RCX file. If you make changes to the RCX file and save it, you must reload it so Synapse can parse it again.

For user gizmos that you want to make readily available in any experiment, place the RCX files in the UserCircuits folder of the Synapse installation directory (typically C:\TDT\Synapse\UserCircuits). Synapse reads this folder and displays the RCX files as gizmos in the Custom category in the Gizmos list so they are always available.

Creating Your Own User Gizmos

User gizmos are designed in RPvdsEx by adding components or macros (pre-made groups of components), linking them together in a logical order, and compiling them as an RCX file.

Prerequisites

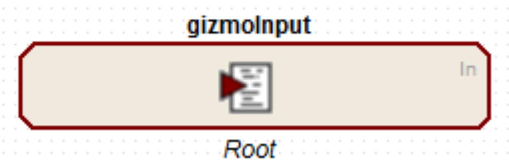
This tutorial assumes basic RPvdsEx knowledge of creating processing chains, working with macros and using parameter tags to read/write values dynamically.

Circuit Requirements

The user gizmo macros are available in the Components > Circuit Macros menu in RPvdsEx. There are three macros specific to Synapse user gizmo circuit design that are available in C:\TDT\RPvdsEx\Macros\Synapse: **gizmoInput**, **gizmoOutput**, **gizmoControl**.

Inputs

Every user gizmo circuit must have at least one gizmoInput macro. If the user gizmo does not require a data source, for example: if you are designing a signal generator to be used as a data source for other gizmos, set the gizmoInput macro Input Role to 'Root'.



GizmoInput Macro

Each user gizmo can receive data from up to four data sources. For each data input into the user gizmo circuit you must add an additional gizmoInput macro and set the **Input Role** to one of 'Input-1' to 'Input-4'. The inputs must be sequential with no gaps in numbering. If the user gizmo requires at least one data source, you do not need a 'Root' gizmoInput macro in your circuit, 'Input-1' takes its place.

For each input data source you specify the allowed data type and channel count range so the Synapse compiler can properly connect it to other gizmos.

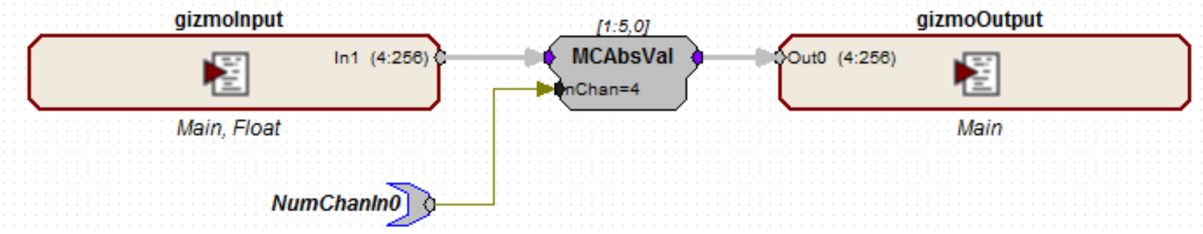
Outputs

A gizmoOutput macro is required if the user gizmo will be a data source for other gizmos. Up to four outputs are allowed in each user gizmo, named 'Output-1' to 'Output-4'. Each output requires its own gizmoOutput macro, where you specify the name, data format and allowed channel count range. The outputs must be sequential with no gaps in numbering.

You also set the output channel dependency. This can be 'Prompt' if the user gizmo is acting as a signal generator, which means the user will select the channel count at designtime, or you can link the channel count to one of the gizmoInput channel counts. For example, a user gizmo that does some custom filtering on a multi-channel signal would likely have an output channel count that matches the input channel count.

If you need to use the channel counts for the inputs/outputs in the processing chain, use specially named parameter tags. These parameter tags must be named **NumChanIn{n}** and **NumChanOut{n}** where n is 0, 1, 2, or 3, one for each of the four possible inputs and four possible outputs. Anything connected to one of these special parameter tags is given the value of the specified channel at compile time.

In the example below, the user gizmo is performing an absolute value operation on a multi-channel floating point data source and making the resulting signal available to other gizmos. It accepts between 4 and 256 floating point channels on the input. Here, the 'nChan' parameter of the MCAbsVal component will be replaced by the number of channels on the input data source when Synapse compiles this circuit into the processing tree. This ensures the MCAbsVal component will always have the correct number of channels.



Custom Absolute Value Gizmo Circuit

User Interface Widgets

If there are parameters in the processing chain that you want to control at designtime or runtime, or that you want to display to the user at runtime, you can specify a user interface widget and attach it to a specific parameter tag in your circuit.

Add a gizmoControl macro for any parameter tag that you want to display a user interface widget for. The gizmoControl macro determines whether this tag is read or write, what type of widget to display, when to display the widget (designtime or runtime), and other configuration options.

The parameter tag name must always be prefixed with "ID_". When Synapse compiles the processing tree, "ID" is replaced by the gizmo name. This allows you to use multiple instances of the user gizmo and prevents naming conflicts.

If any user interface widgets are specified to show at designtime, they will appear on the Control tab in Synapse. Any that show at runtime will have their own tab in the runtime interface. The controls will be organized alphabetically by parameter tag name on the runtime screen based on window size.

Removing Unused Components

It is important to keep circuit design in your user gizmos as efficient as possible. If you're unable to us multi-channel components and must instead use an iterate box in your circuit, you can dynamically remove unused components inside iterate loops by naming them "KILL~{x}".

Two-Sample Delays

Like all gizmos, user gizmos add a two sample delay to the processing path. This is particularly important to keep in mind for tasks where timing is critical. In cases where you have more than one signal or processing path, RPvdsEx delay components (such as: SampDelay, MCDelay) can be placed in a user gizmo to synchronize the paths.

MATLAB/Python Access

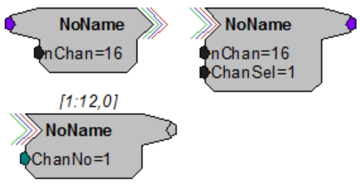
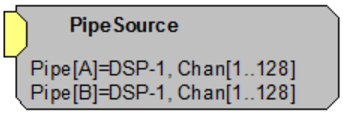
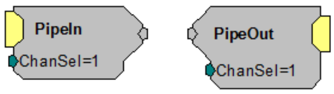
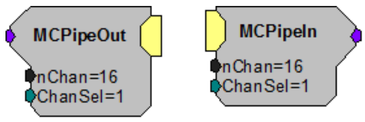
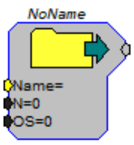
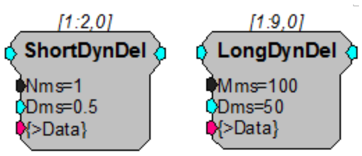
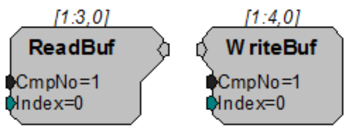
SynapseAPI can also be used to read and write parameter tags in the user gizmo circuit. The parameter tag name must always be prefixed with "ID_" to avoid naming collisions when multiple instances of the same user gizmo are used on the same device.

See the SynapseAPI Manual for more information.

User Gizmo Do's and Don'ts

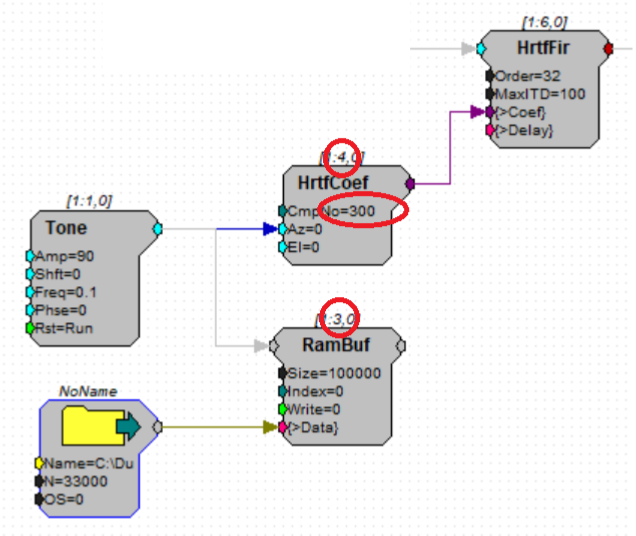
The following list of RPvdsEx components are not available in user gizmo circuits. Some may have alternatives.

Unsupported RPvdsEx Components

Component Name	Graphical Representation	Synapse Alternative
zHop Components		Use the gizmoInput / gizmoOutput macros to share signals between gizmos
MCzHopOut MCzHopIn		The standard timing zHopIns from OpenEx (iTime, Reset and Enable) can be used in the user gizmo circuit
PipeSource		Use the gizmoInput macros to share signals between gizmos
PipeOut PipeIn		
MCPipeOut MCPipeIn		
DSP Assign		Use DSP assignment option in gizmoInput macro to force a user gizmo to run on a specific DSP.
SourceFile		Use SynapseAPI to load a buffer instead.
ShortDynDelay LongDynDelay		None
ReadBuf WriteBuf		None

HRTF Support

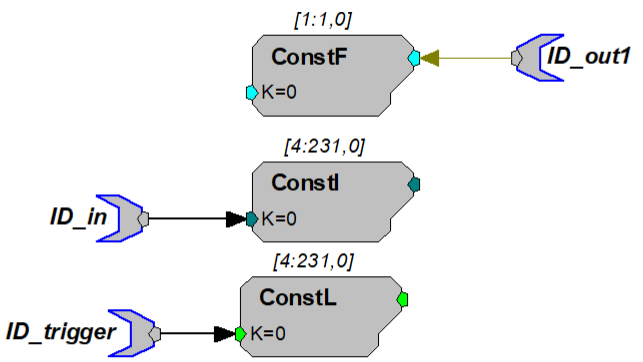
Use a circuit construct like the one shown below to force the compiler to place the RamBuf component just before the HrtfCoef component in the processing order. Verify it is properly ordered by clicking the compile button in RPvdsEx. Then specify a CmpNo = 300 on the HrtfCoef component.



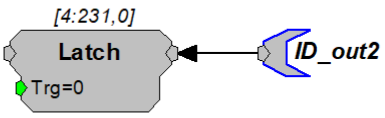
Parameter Tags

Parameter tags must be attached to a port that is typed (float, integer, logic). Do not connect a control tag to a gray non-typed port.

good

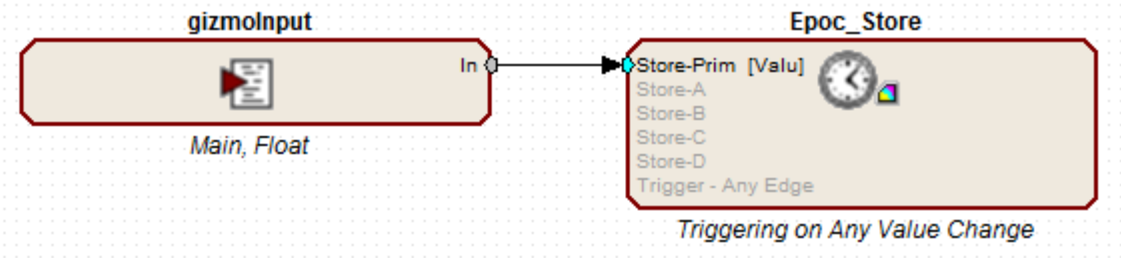


bad



Legacy OpenEx macro support

The standard OpenEx macros can be used in user gizmos. All of the timing structures you need are included in the **gizmoInput** macro. The SpikePac macros are NOT supported in user gizmo - use their corresponding TDT Gizmo replacements instead.



Artifact Blocker

Common Use Cases

Suppress artifacts associated with triggered events. Includes gate timing parameters for control of gate shape. Use this gizmo to remove large artifacts during events like electrical stimulation or motion artifact.

Data Stored	
Epoc (optional)	Timestamps for each artifact
Outputs	
Main	Single or multi-channel floating point signal with artifact removed
Gate Control	Time-locked logic signal reflecting artifact gate

Gizmo Help Slides

Quick Help Slide 1

Artifact Blocker — Suppress recording artifacts using controllable signal gate. Titrate gating tightly around artifact to clean up online signals

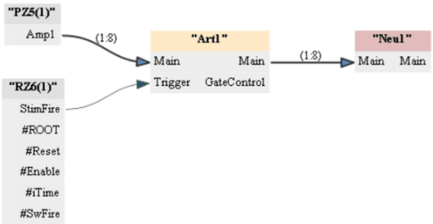
Neural recording with electrical stimulation

Recording in an MRI

Startle response suppression

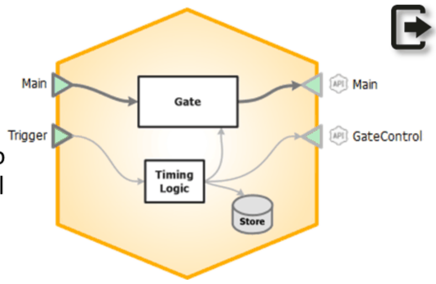
Transducer input processing

Suppress gradient switching artifact in MRI using timing signal from magnet



Main: Input fed from neural amplifier or other high sensitivity signal

Trigger: Connected to controlling logic signal that is time-locked to artifact



Main: Output typically passed to processing or visualization gizmo

GateControl: Time-locked logic signal reflecting gate control

Quick Help Slide 2

Artifact Blocker

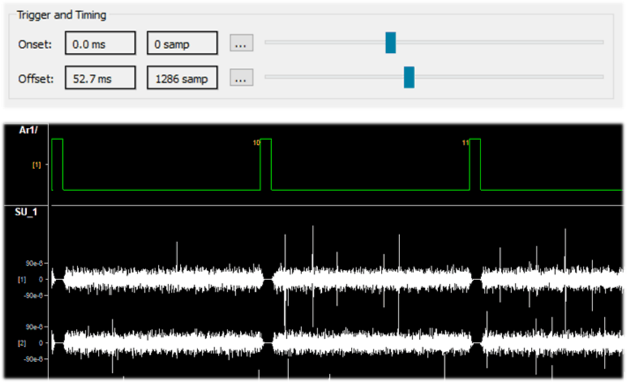
User Interface, API, Data Stored

Dynamic control of gate timing allows titration of blocking at run-time.

Onset — Gate onset control in milliseconds {Read/Write}

Offset — Gate offset control in milliseconds {Read/Write}

Ar1/ — Gate Timing signal: full epoch capturing gate onset and offset {Epoch}



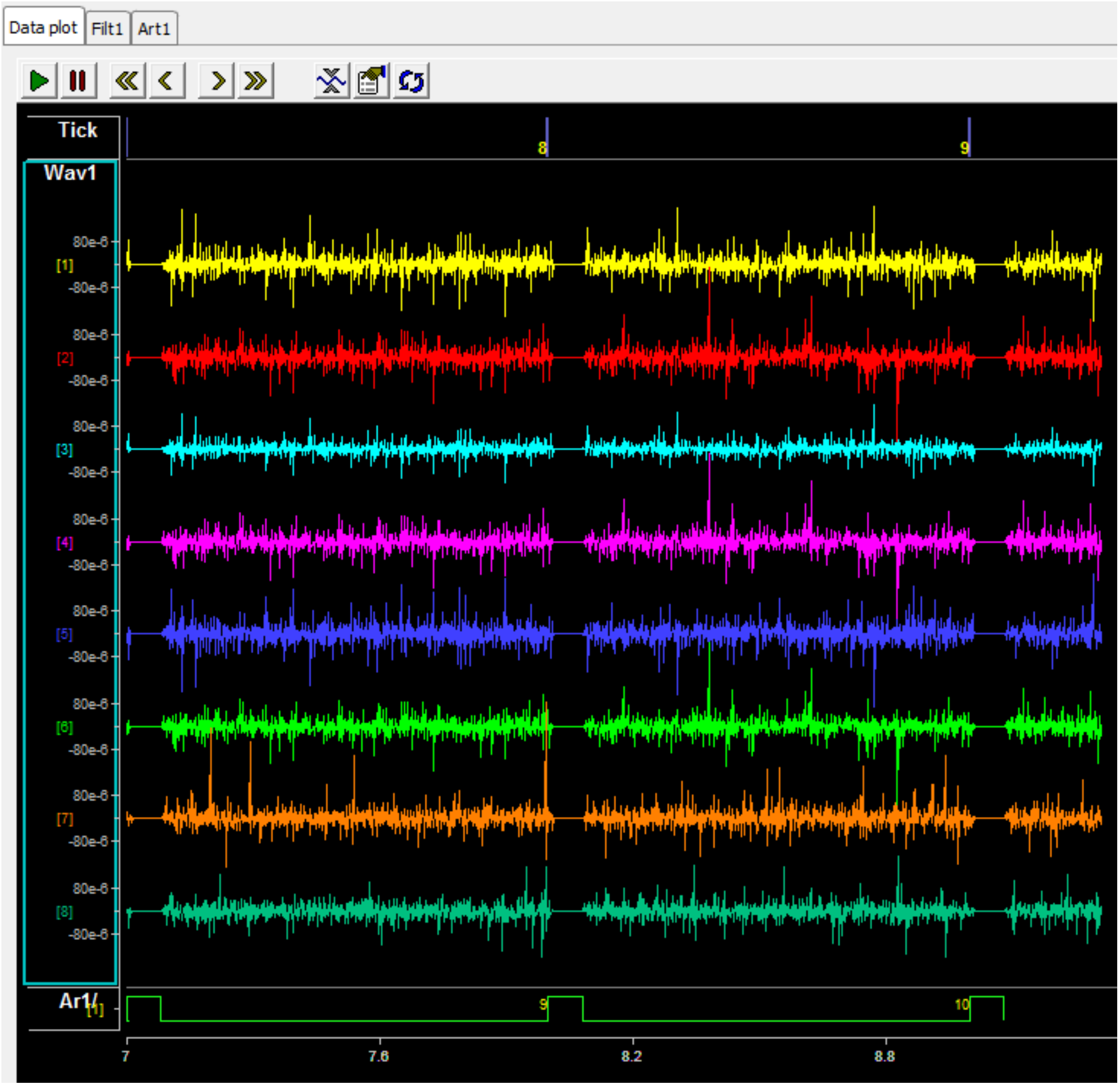
Reference

The Artifact Blocker gizmo zeros a signal relative to a trigger, blocking stimulus artifacts in recorded data associated with a triggered event. Timing logic can be stored and/or used as a source for other gizmos.

The Artifact Blocker Runtime Interface

Runtime Plot

If you choose to save gate timing, a plot showing the timing of the gate is added to the runtime window for visualization.



Runtime Plots include Artifact Timing Data

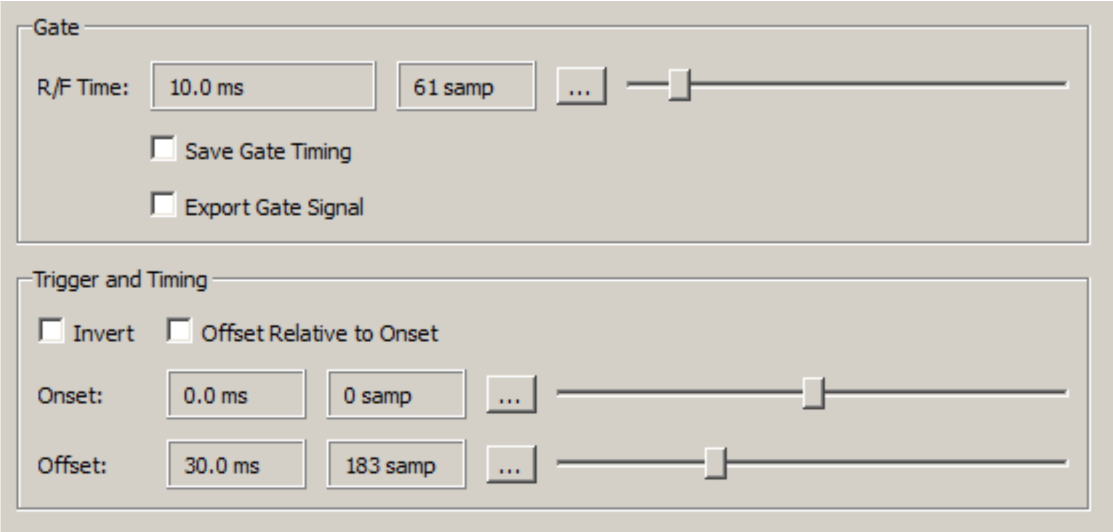
The main runtime plots show where artifact rejection has been applied to the neural signals.

Artifact Blocker Tab

Artifact Blocker Runtime Tab

The Artifact Blocker tab has sliders to dynamically adjust the gate onset and offset timing at runtime. If the onset is less than zero, the incoming signal is delayed by that many samples in order to synchronize with the trigger.

Artifact Blocker Configuration Options



Trigger Options

Gate

The artifact blocker uses a cosine-squared gate. The rise/fall time (**R/F Time**) represents the amount of time it takes to reduce the signal by 90%, and to increase it back to 90% of its final value.

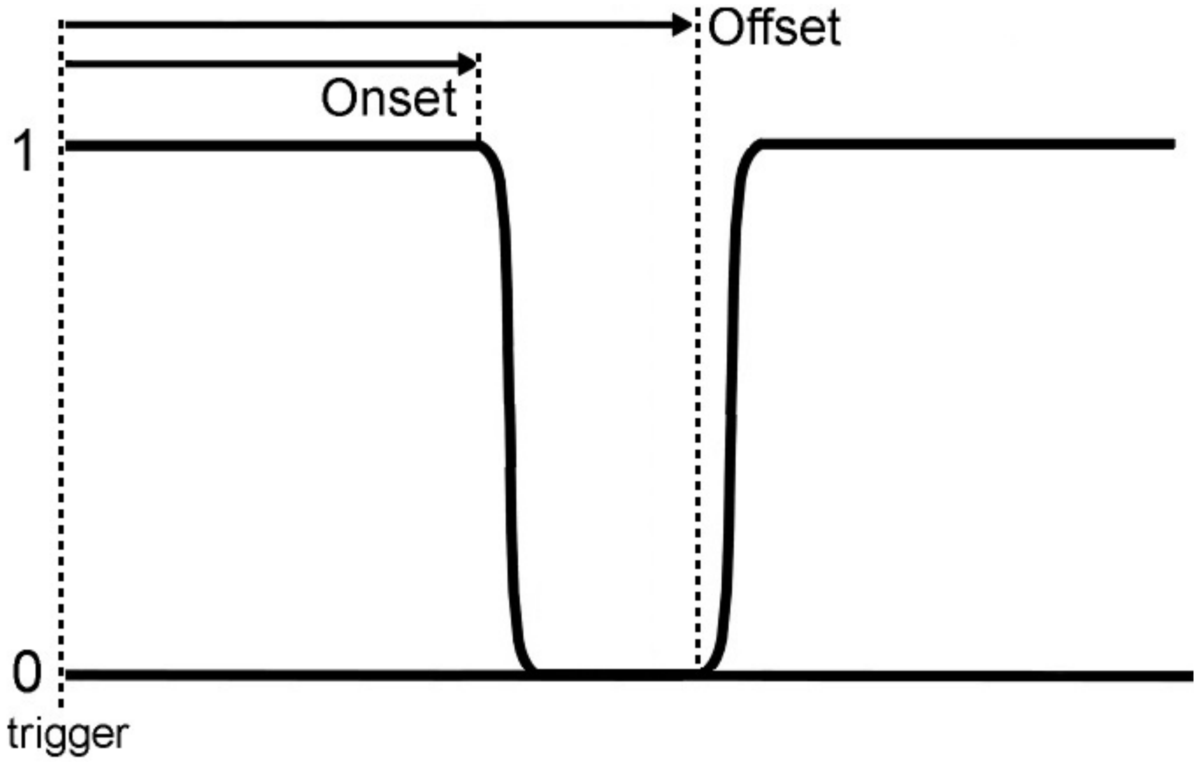
If the specified rise/fall time is less than four samples, a square edge is used which immediately scales the signals by 0 when the trigger onset occurs.

The **Save Gate Timing** stores the timing signal in the data tank. The **Export Gate Signal** check box makes the gated timing signal available as an output.

Trigger and Timing

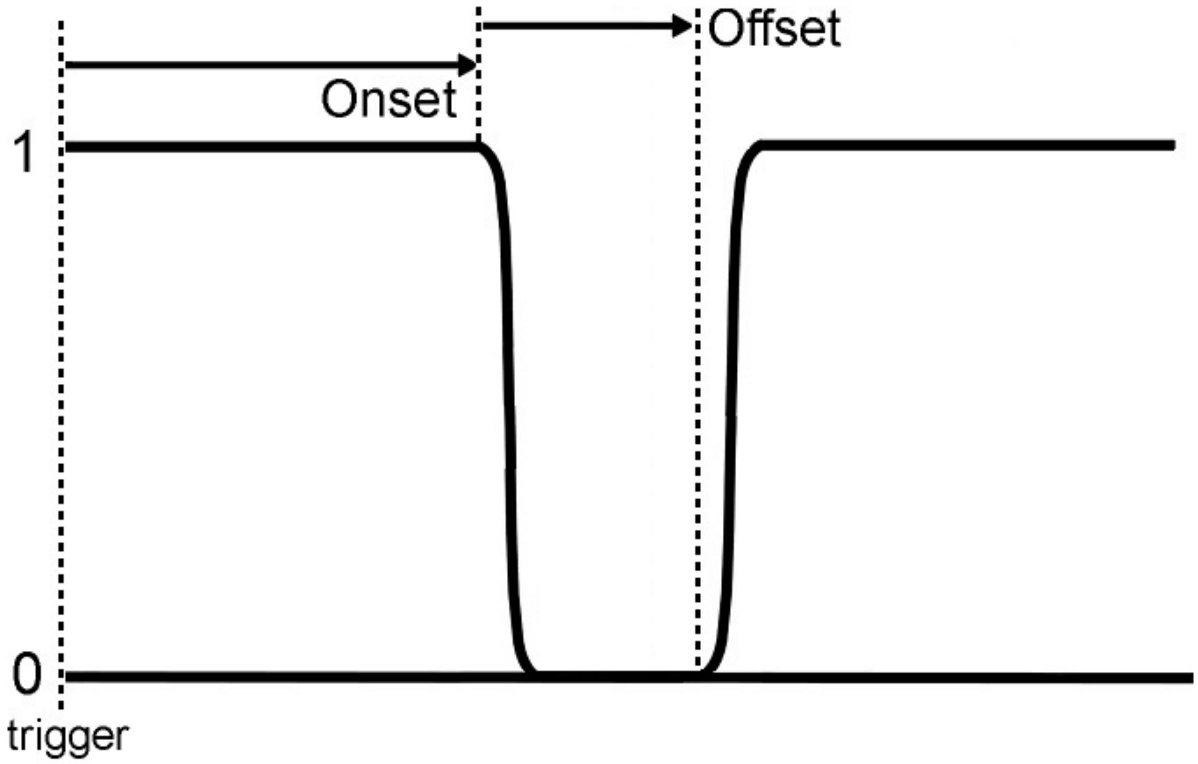
Check the **Invert** box to reverse the polarity of the trigger input.

Set the default **Onset** and **Offset** of the gate relative to the onset of the trigger input. By default the offset is also relative to the trigger, as shown in the gate depiction below.



Default Timing Diagram

If **Offset Relative to Onset** is selected, the gate signal timing looks like this:



Offset Relative to Onset Timing Diagram

Audio Stimulation

Common Use Cases

Generate fully customizable tone, noise, and other audio stimulation types. Use this gizmo for audio neurophysiology, stimulus-response protocols, hearing screening protocols, and psychoacoustics.

Data Stored	
Epoc (optional)	Parameter values when triggered
Stream (optional)	Raw stimulus waveform
Outputs	
Stim	Stimulus waveform, single channel floating point
StimSync	Logic signal when audio stimulation is active
Par Output	Full parameter stream

Gizmo Help Slides

Quick Help Slide 1

Audio Stimulation — Generate fully customizable tone, noise, and other audio stimulation types

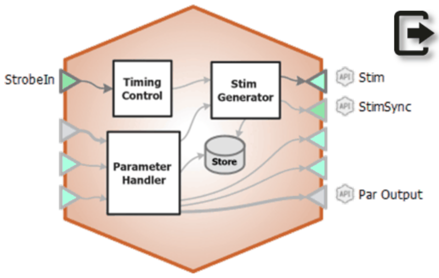
Stimulus-response protocols

Hearing screening protocols

Psychoacoustic experiments

RZ6 processor #Reset tied to the Audio Stim gizmo allows manual control of stimulus presentation. Sound output sent to DAC1 for speaker output and to digital IO for external TTL communication with peripheral devices.

StrobeIn: Optional input to control presentation timing



Stim: Single-channel stimulation signal

StimSync: Logic signal that is high during a stimulus presentation

ParOut: Parameter values for the given presentation

Quick Help Slide 2

Audio Stimulation

User Interface, API, Data Stored

Manually strobe and mute stimuli presentations. Change waveform parameters during runtime.

Mute — Turn the stimulus on & off during a presentation {Read/Write}

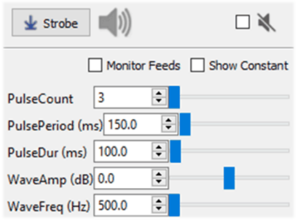
ParameterList — Array of all stimulus parameters {Read}

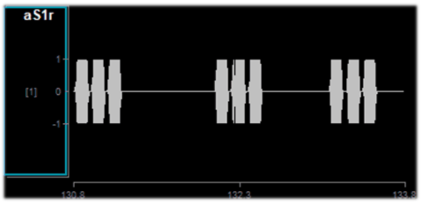
StimActive — Logic high during a stimulus presentation {Read}

Strobe — In Manual mode, strobe the gizmo to give a stimulus presentation {Write}

aS1r — Continuous audio output stream {Stream}

aS1p — List of all stimulus parameters stored when stim occurs (optional) {Strobe}





Quick Help Slide 3

Audio Stimulation

Parameter Table

This gizmo has a Parameter Table to dynamically control values its parameters from other gizmos and during runtime

Parameter Tables provide the user dynamic control over parameter values. Each one of the Modes will change how the user can control and interact with the parameter.

Param In – Dynamically control parameter values from Parameter Sequencer or Parameter Manifold

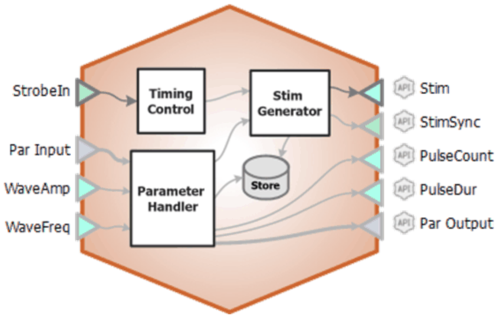
Constant – The value is immutable at runtime

Scalar In – Pass a scalar value from another gizmo for dynamic real-time control

Widget – Creates a widget (slider) on the user interface that is controlled manually by the user or through the API.

Scalar Out (Scout) – Makes this value available as a gizmo output to connect to another gizmo input in real-time

Epoc – Save the parameter values used for each stimulus presentation



Changing the Mode of parameters will change the input and output options on the gizmo.

All parameters can be read through API. Only parameters in Widget mode can be written to.

All parameters are connected to the “Par Output” gizmo output link

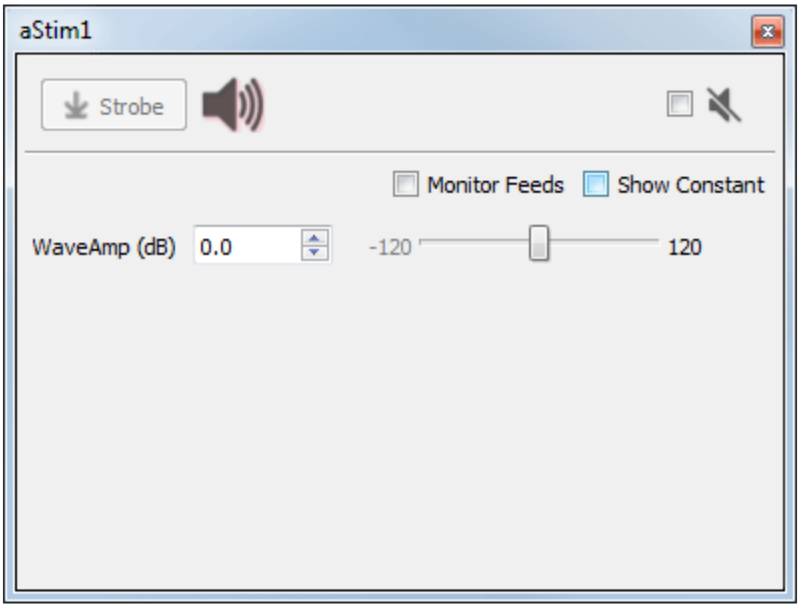
Audio Stim Parameters: Show All

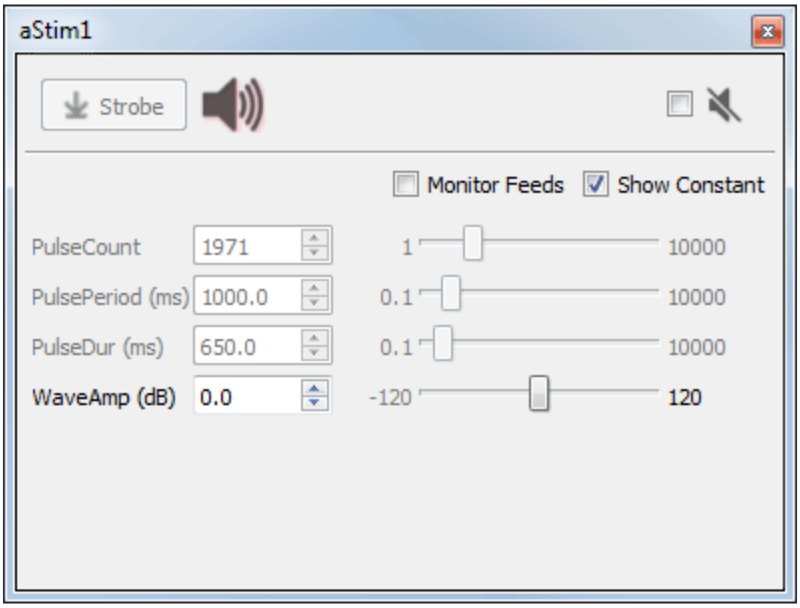
	Name	Mode	Value	Jit(%)	Min	Max	Epoc	ID	Auto ID
2	PulseCount	Constant	1	0.0	1	10000	None	Pcnt	☑
3	PulsePeriod (ms)	Widget	100.0	0.0	0.1	10000.0	None	Pper	☑
4	PulseDur (ms)	Param In	30.0	0.0	0.1	10000.0	None	Pdur	☑
5	WaveAmp (dB)	Scalar In-1	0.0	0.0	-120.0	120.0	None	Wamp	☑
6	WaveFreq (Hz)	Scalar In-2	1000.0	0.0	0.1	100000.0	None	Wfrq	☑

Reference

Audio stimulation waveforms may be comprised of tones, noise, sawtooth or square waves that can vary in duration, level, and more. The overall stimulation duration can be set by a fixed duration, based on a strobe or based on pulse count. The gizmo provides static or runtime control of stimulus parameters and can input parameters from a Parameter Sequencer gizmo. The audio stimulation gizmo includes options to store individual parameters, the parameter list, and raw waveform. A timing pulse can also be output to synchronize data collection.

Audio Stimulation Runtime Interface





Two Versions of the Audio Stimulation Runtime Tab

An aStim1 control tab is added at runtime. Parameters that can be controlled dynamically are shown in black (active). You can enter a value in the field, use up and down arrows, or drag a slider to modify to parameter value. You can show only the elements you need or hide the entire control. The illustrations above show two versions of the floated tab, one with only the runtime widget-controlled parameter shown and one with all the parameters shown.

Click and release the **Strobe Button** to trigger a manual strobe pulse.

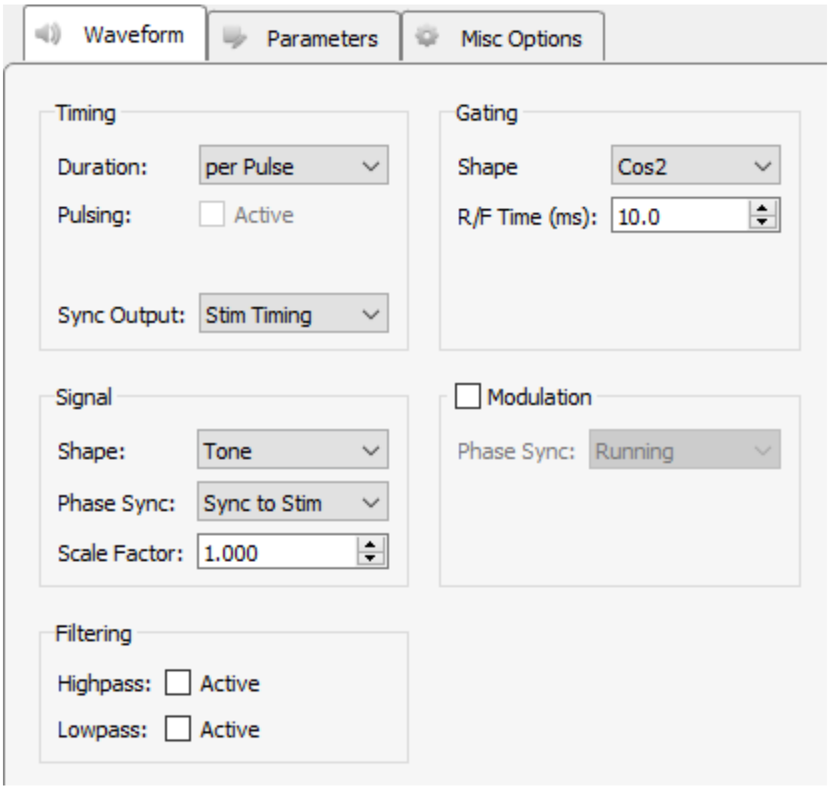
Select the **Mute Button** check box to zero the stimulus signal.

Select the **Monitor Feeds** check box to show stimulus parameters controlled by an input signal. Also adds an Override column and check box to the left. Select the **Override** check box to adjust the parameter value manually instead of using the input signal.

Select the **Show Constant** check box to display values for parameters set to Constant. They will appear gray.

Audio Stimulation Configuration Options

Waveform Tab

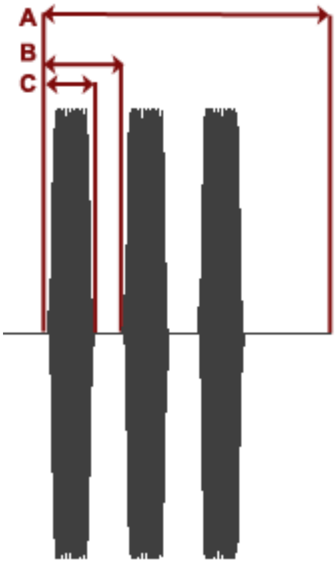


Waveform Tab

Timing

Set the **Duration** of the stimulus waveform: per Pulse, per Parameter, or per Strobe. Descriptions are below.

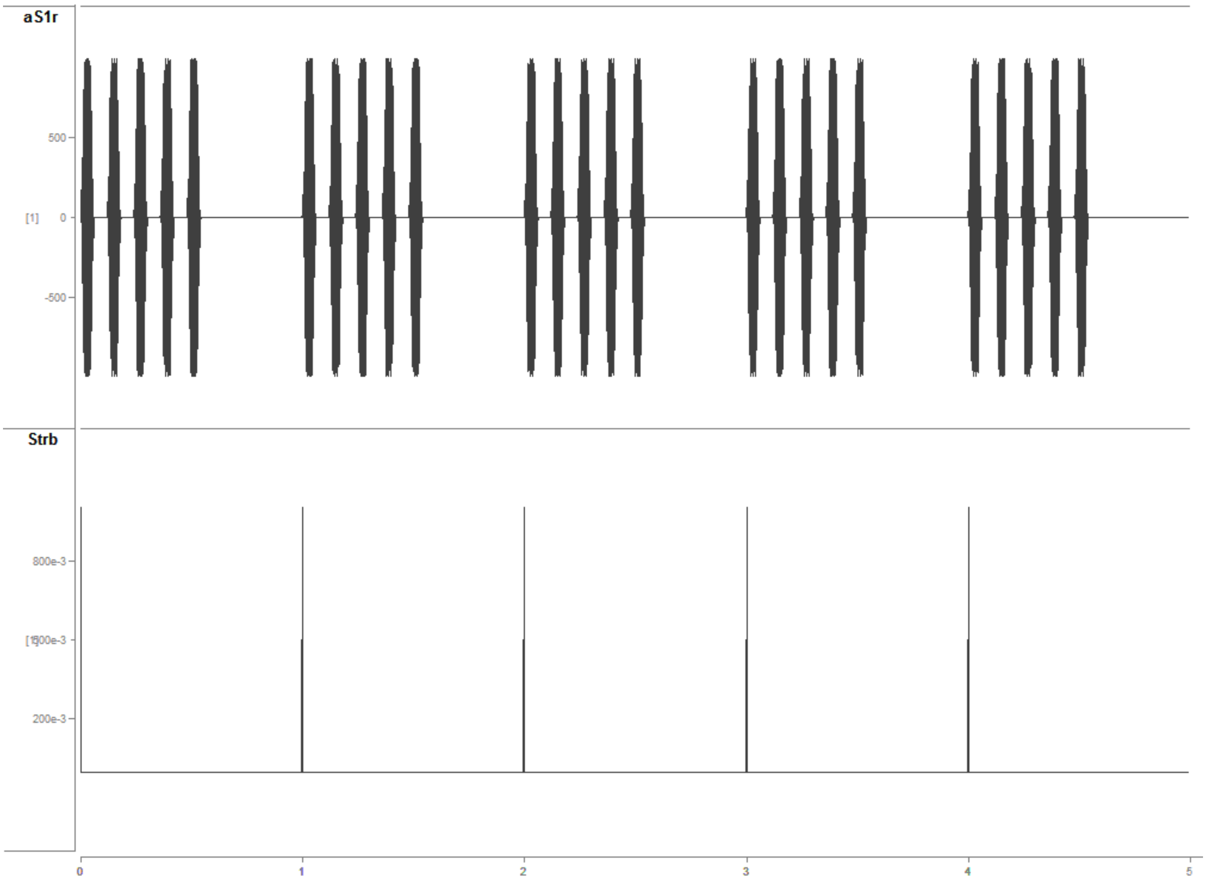
1. **per Pulse** enables you to set a pulse count, duration, and period (in the parameter table) relative to a pulse input to the gizmo's Main input. Typically this is your #SwFire (stimulus sweep fire line) or similar repeating pulse. The diagram below provides a quick visual guide to the three parameters of the stimulus.



Stimulus Tone per Pulse

No matter which method is used to design the stimulus, the next trigger begins a new stimulus. This ends the previous stimulus whether or not the stimulus duration or pulse count has been reached. Before your experiment, be sure to preview your stimulus to ensure it is working as expected.

The plot below shows a tone pulse train, triggered by a sweep fire line (#SwFire)--a typical Synapse timing element available on RZ devices. It fires once every second.



Stimulus Waveform Shown Above with Timing Pulse Below

2. **per Parameter** is similar to **per Pulse** in that it also enables the pulse count and period parameters (in the parameter table). However, the duration parameter is also enabled, so you can define the duration as a time period via the Parameters Table tab where you have the additional options to:

    a. Set a constant value

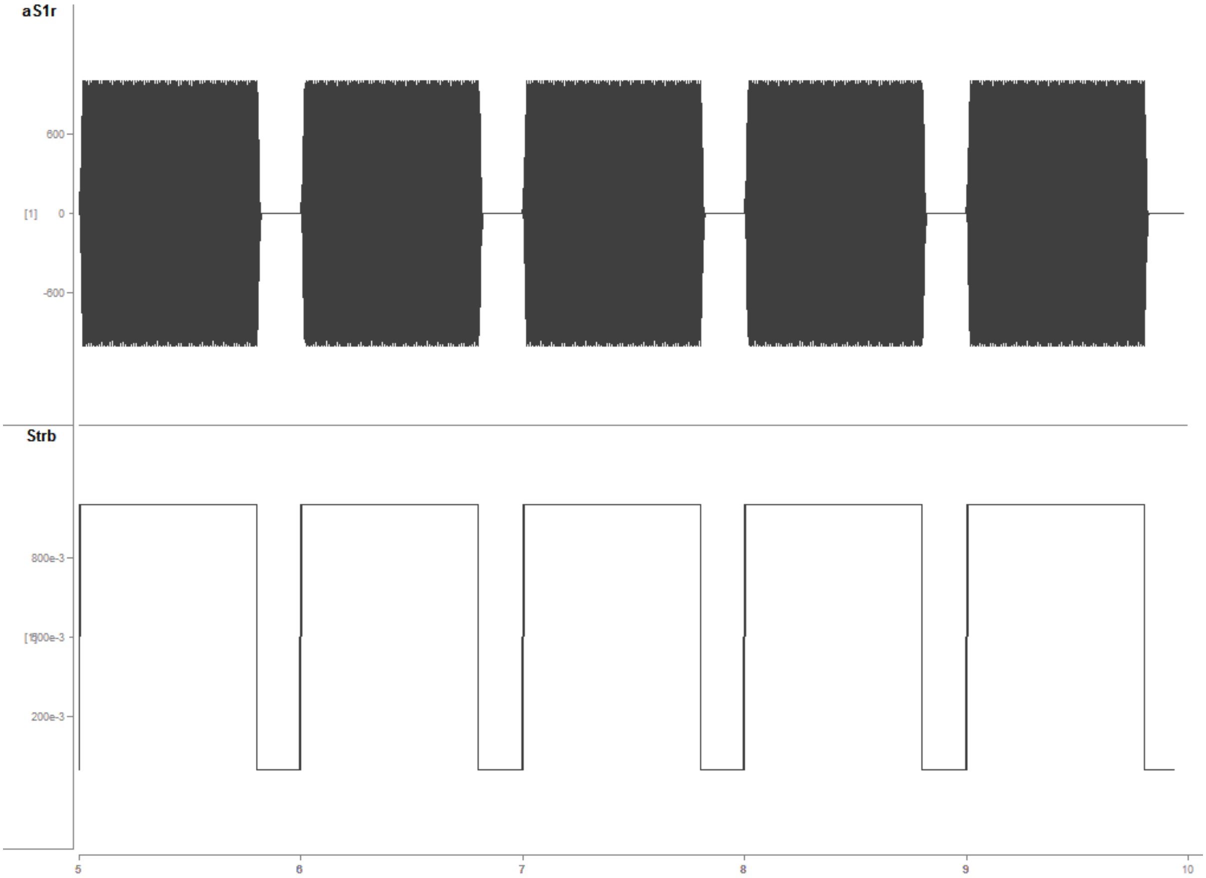
    b. Use a runtime widget

    c. Use a parameter input from a parameter sequencer

    d. Use an input from a scalar input line

> **Tip**
>
> See Using Parameters for more information on controlling parameter tables.

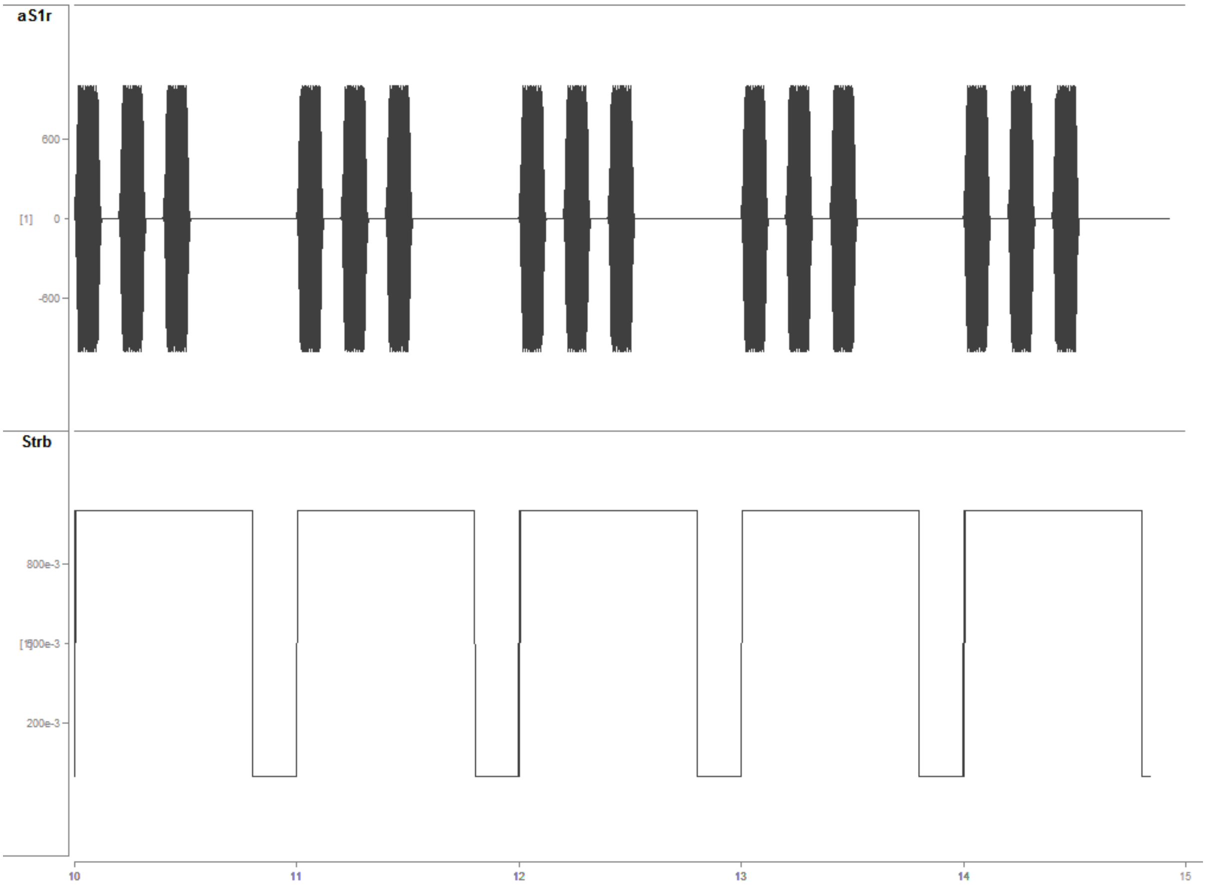
3. **per Strobe** also uses the gizmo's Main input to trigger the signal, however, because no pulse parameters are applied the signal matches the duration of the Main input source, typically a strobe input.



Stimulus Timing Per Strobe

Pulsing Active check box

When selected, pulse duration and pulse count parameters are enabled in the parameter table and the stimulus is triggered when the strobe goes high, the pulse parameters are then followed and the stimulus ends with the pulse count is met or the strobe goes low. The next stimulus is triggered by the next strobe input.



Stimulus Timing Per Strobe with Pulsing Active

Gating

Shape

Choose the type of gate to apply to the signal. Gates serve to attenuate the signal during the onset and offset of the signal, increasing or decreasing in intensity, for the purpose of removing onset/offset related artifacts from this signal.

R/F Time (ms)

Defines the length of time over which the gate is applied, therefore, the length of time in which the signal goes from 0 to full signal strength or visa-versa.

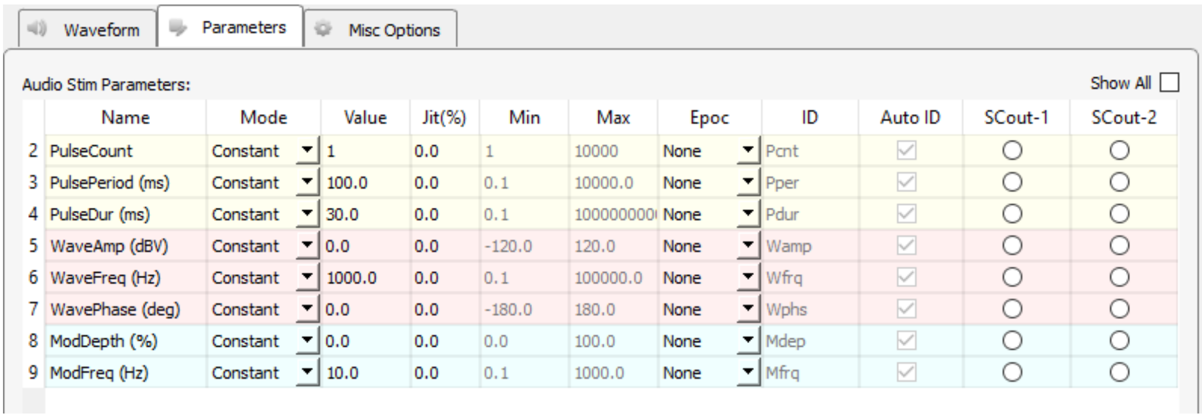
Signal

Select the desired waveform shape and related properties. Select the Modulation check box to add amplitude modulation and select whether to synchronize the phase of the modulation waveform. Note that when Phase Sync is 'Sync to Stim' or 'Sync to 'Pulse', a WavePhase parameter is available in the Parameters Tab. It is limited to the range [-179.9, 179.9], even if the table says -180 or 180.

Filtering

Filters can be applied to the output signal. Useful for band-limited noise presentation.

Parameters Tab



Audio Stimulation Parameters Tab

Audio Stimulation Parameters

The table lists signal parameters relevant to configuring a stimulus. Each row represents a parameter and rows are shown or hidden in response to selections made on the Waveform tab. Use the **Show All** check box to display hidden rows.

> **Note**
>
> The 'WaveAmp (dBV)' parameter is **not** dBSPL, which is a common unit for measuring sound levels relative to human hearing. The dB units are relative to 1 V, which means requesting 0 dB amplitude will produce a 1 Volt peak-peak waveform, and 20 dB will produce a 10 Volt waveform (which is the maximum output of the RX and RZ DACs). Going beyond 20 dB without some sort of conversion downstream will clip and distort your sound output.
>
> Please contact support@tdt.com if you have any questions.

> **Tip**
>
> See Using Parameters for more information on using the parameters table.

Mode

In the Mode column, you can choose to make individual variable Constant, controlled by a runtime Widget, fed by a parameter input line (from Parameter Sequencer gizmo) or controlled by a Scalar Input line.

Value Columns

Enter values in the Value, Jit% (Jitter), Min, and Max columns to set the Constant value or to set the initial value and limits when parameters are dynamically controlled. In Widget mode, the Min and Max set the Widget limits.

Epoc

In the Epoc column, you can choose to save the individual parameter value on stimulus or pulse onset.

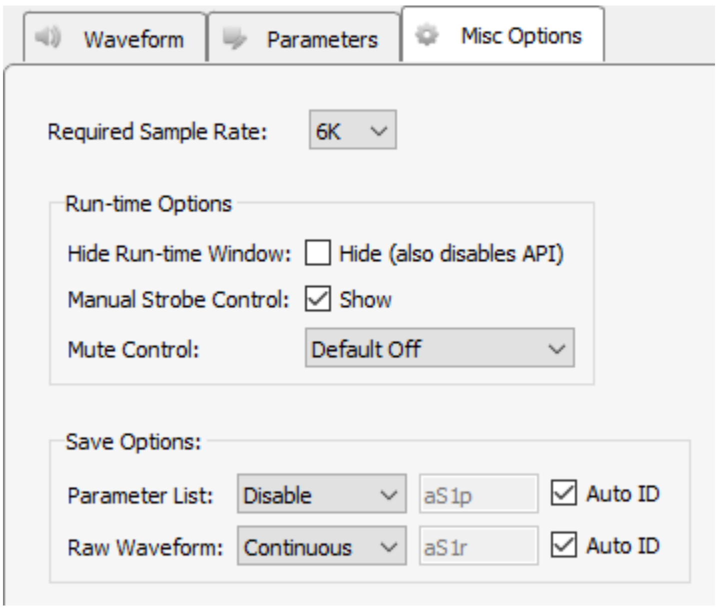
ID and Auto ID check box

Synapse automatically generates a store name for each parameter. TDT recommends using Auto ID to ensure no store names are duplicated. A "/" is appended to the name to indicate that the full epoc (onset and offset timestamp) is stored.

SCout-1 and SCout-2

Select the radio button in the desired row to feed the parameter to an output signal on the gizmo.

Misc Options Tab



Misc Options Tab

Required Sample Rate

The minimum rate required. Synapse looks through the entire experiment and your Rig and sets the sample rate according to this and other limiting factors.

Hide Run-Time Windows

By default a runtime tab is added in Preview or Record mode. The contents of the tab are defined with configuration options on the General and Parameter options tab. Select the check box to hide the runtime tab.

Manual Strobe Control

When selected, a manual strobe control is added to the runtime UI.

Mute Control

Mute allows you to temporarily mute the stimulus during runtime. You can choose to hide or show the control and, if shown, set the default start state.

Parameter List

Select whether to store the value of all parameters, at each stimulus or pulse onset. This generates a multi-channel list of scalar values. The channels map directly to the rows of the parameter table on the Parameters tab. By default, some parameters are hidden in the table, but values are stored for all parameters.

Raw Waveform

Select whether to store a copy of the raw stimulus waveform. You can choose to store continuously or only when the stimulus is active.

Auto ID field and check box

A store name is generated automatically. To use your own store name, clear the **Auto ID** check box.

Box Spike Sorting

Common Use Cases

Real-time filtering, spike detection, and discrimination of neural signals using time-voltage windows. Use this gizmo to sort neuronal spikes on individual channels using time-voltage discrimination windows

Data Stored	
Snippets (optional)	Timestamped spike waveforms
Stream (optional)	Plot decimated waveforms
Outputs	
Main	Filtered, multi-channel floating point signal
Sort Codes	Multi-channel integer signal containing compressed sort codes

Gizmo Help Slides

Quick Help Slide 1

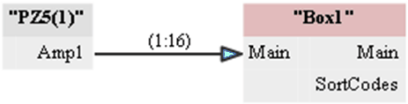
Box Spike Sorting — Apply real-time filtering, spike detection, and discrimination of neural signals using time-voltage windows

Neural recording with action potentials

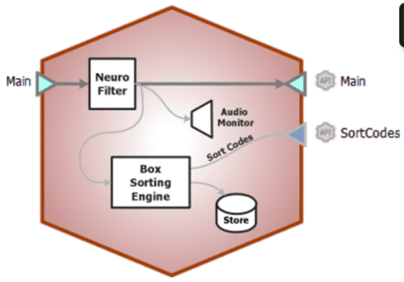
Acute neurophysiology

Chronic behavior experiments

16-Channel Box spike sorting from raw amplifier data stream



Main: Input fed from neural amplifier



Main: Filtered signal output. Typically unused because data stream can be saved directly in gizmo

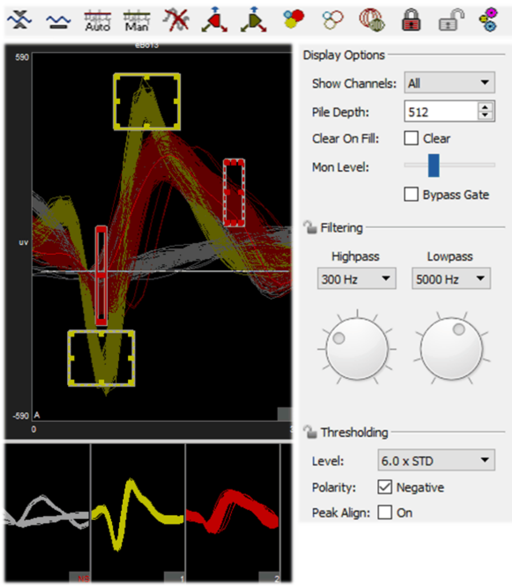
SortCodes: Multi-channel integer values containing sort code information. Typically goes to Selector or Sort Binner for further online processing

Quick Help Slide 2

Box Spike Sorting

User Interface, API, Data Stored

Dynamic control of filters, thresholding, and clustering allows for advanced clustering of thresholded snippets.



FreqHP/FreqLP — Corner frequency control of filters {Read/Write}

ThreshLock — Lock the threshold level (per channel) {Write}

ThreshMode— Switch between manual and dynamic thresholding (per channel) {Read/Write}

Threshold — Change the threshold level (per channel) {Read/Write}

pBo1 — Filtered plot-decimated data stream {Stream}

eBo1 — Captured spike snippets with assigned sort codes (if Hardware Sort enabled) {Snippets}

Reference

The Box Spike Sorting gizmo performs filtering, thresholding and online time- voltage spike sorting and storage on multi-channel neural signals at sampling rates up to 50 kHz.

Data Storage

This gizmo generates two types of data for storage: snippet data (includes timestamp, short waveform, and sort code) and plot decimated data streams. The stream data generated by this gizmo is a highly decimated version of the waveforms that keeps local maximum and minimum values of the filtered signals, which makes it ideal for visualizing high frequency spike activity on a computer monitor with a fixed number of pixels.

In plots and in the data tank, each type of data is designated with a prefix: 'e' for snippets and 'p' for streams. You can opt to save only snippets or to disable storage in the gizmo's configuration settings. The sort codes can be configured as an output to be used in other gizmos.

Threshold Detection

At runtime, candidate spikes are detected based on a calculation of the deviation of a waveform from its RMS. By default, the timestamp and position of the waveform in the snippet is dependent on the time of the threshold crossing for the signal. An alternative setting allows waveform timestamp and positioning to be determined by the waveform's highest peak, aligning snippets to their respective peaks. By default, detection is automated and you can make adjustments in the threshold control plot in the runtime window.

Spike Sorting

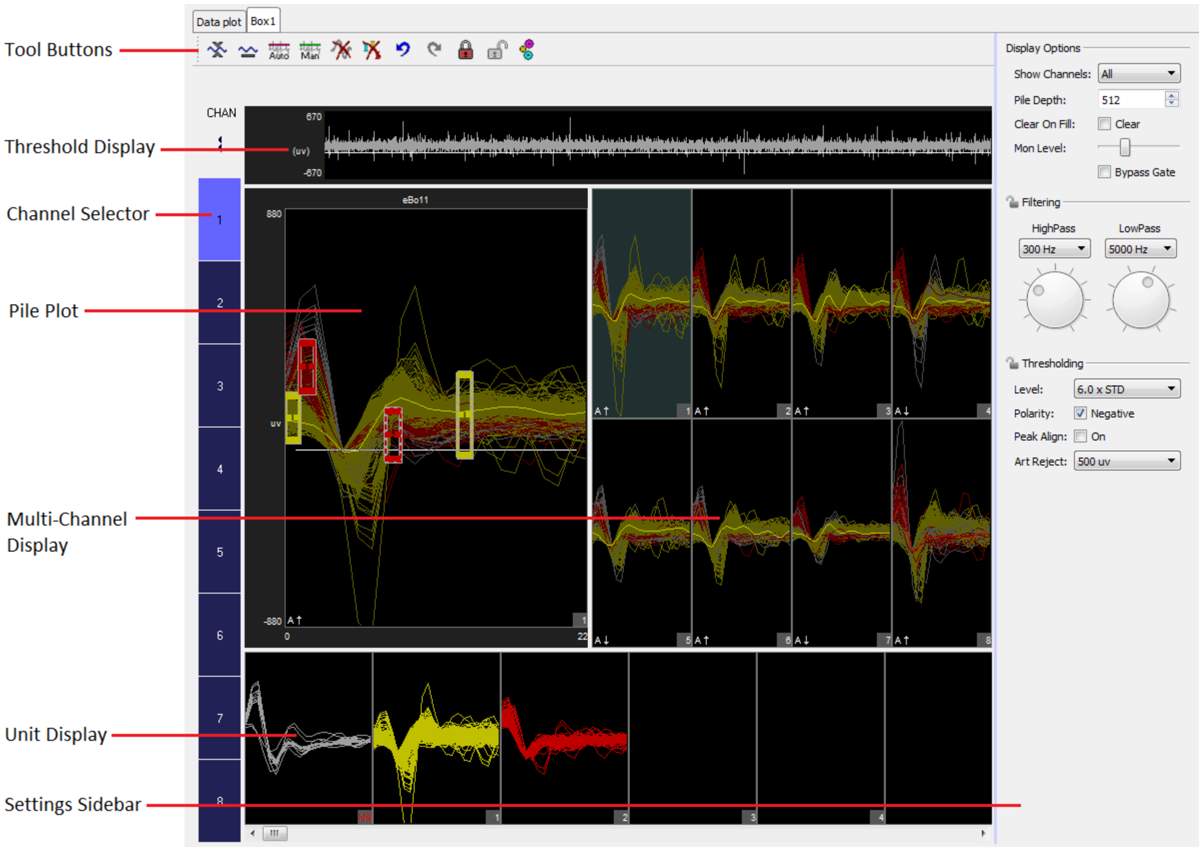
A runtime window tab offers manual sorting using time-voltage box pairs to classify potential units among candidate waveforms. When satisfied with the sorts for all channels, the user can choose to apply **Hardware Sorts**. The sorting parameters are sent to the hardware and sort codes will be applied to new data as it is acquired in real-time. This toolbar button must be 'pressed' for online sorting to take place on the hardware.

The Runtime Interface

Runtime Plot

Streamed waveform and Snippet plots are added to the runtime window for visualization.

Box Spike Sorting Tab



Box Spike Sorting Window

The runtime window includes:

Tool Buttons	Performs actions that are global to all channels.
Threshold Display	Displays the plot decimated waveform of the currently selected channel and the threshold marker. When automatic threshold tracking is active the threshold bar is locked.
Channel Selector	Selects the active channel and indicates channel status. Gray indicates the channel is locked and sorting parameters can't be changed.
Pile Plot	Displays candidate spikes for the active channel. Indicators in the bottom left corner denote scaling and threshold tracking states ('A' for automatic, 'M' for manual). Hold down the **Ctrl** key and double-click to add time-voltage windows.
Multi-Channel Display	Displays a pile plot for each channel. The channel number is shown in the bottom right corner of each subplot and new waveforms are highlighted as they are added to the plot. Clicking a subplot makes that channel the active channel for other plots on the tab. Indicators in the bottom left corner denote scaling and threshold tracking states.
Unit Display	Displays a single channel of candidate waveforms by unit - each plot displays all waveforms classified with a single sort code.
Settings Sidebar	Includes settings for display options, filtering, and threshold settings.

Simple Zoom

You can zoom any plot to see more or less detail without affecting the actual data.

To change the zoom level, hold down the **Shift** key and click-and-drag the pointer up or down.

To reset the zoom level, hold down the **Shift** key and double-click within the display area.

Display Scale

To make it easier to see waveform shapes for channels with lower magnitude, you may scale individual channels manually or normalize all channels to fit to a similar scale, all without altering the data being stored.

To normalize all channels, click the **Auto Scale** button in the toolbar and choose to normalize the display. Each channel is scaled individually to fit around 80% of the signal's vertical size in each plot. An up or down arrow is displayed in the bottom left corner of the plot or subplot to indicate whether the display has been scaled up or down. This does not change the scale of the feature space.

To adjust the scale of a single channel, press and hold down the **Ctrl** key, and click-and-drag the mouse up or down in the multi-channel display. While adjusting the display scale, the numeric value in the lower right corner of the channel plot indicates the new scale value.

To reset the scale for all channels, click the **Reset Base Scale** button. This does not remove any zoom applied to a plot.

To return a single channel to its base scale, right-click the desired channel and select **Reset Scaling** from the menu.

Settings Sidebar

Display Options	Description
Show Channels	Select the number of channels to display in the Multi-Channel Display.
Pile Depth	Enter a number to set the maximum number of events displayed in pile plots. The oldest waveform traces are removed as new events are added.
Clear on fill	Select the check box to refresh plots, clearing all traces for a given channel whenever the pile depth is reached on that channel.
Mon Level	Slide the indicator to adjust the level of the audio monitor output, when enabled.
Bypass Gate	A noise gate on the audio monitor removes background noises so only the spikes are heard. Select this check box to turn off the noise gate.

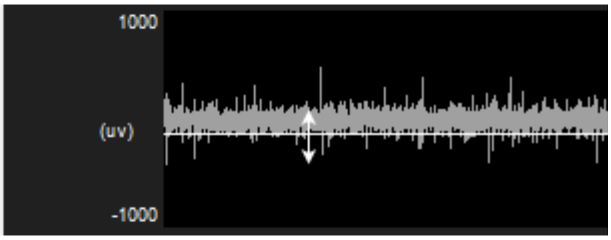
Filtering Options	Description
HighPass/LowPass	Set the highpass and lowpass digital filter settings. The filter is applied to the data before threhsolding, sorting, or visualization

Thresholding Options	Description
Level	Set the automatic threshold level for spike detection, in number of standard deviations from the baseline (smoothed over a 3 second window)
Polarity	Set automatic threshold search polarity, either positive or negative
Peak Align	If enabled, aligns spikes according to their peak values, altering the timestamp of the snippet
Art Reject	When artifact rejection is enabled in the configuration options, sets the artifact rejection level in microvolts. If any sample of the candidate waveform is above this level, the waveform is ignored

Threshold Control

Click the **Auto Threshold** button to initiate automatic threshold tracking on all unlocked channels. If Auto Thresholding is enabled in the designtime interface, real-time tracking will begin on all channels, otherwise the channels will remain in manual threshold mode and the threshold will be set based on a one-time calculation using the current window data and the **Thresholding Level** and **Polarity** settings.

Click the **Manual Threshold** button to enable manual thresholding on all unlocked channels. In manual threshold mode, the threshold bar may be adjusted by clicking and dragging the white bar in the threshold display window (shown below) or in the pile plot.



Threshold Display in Manual Mode

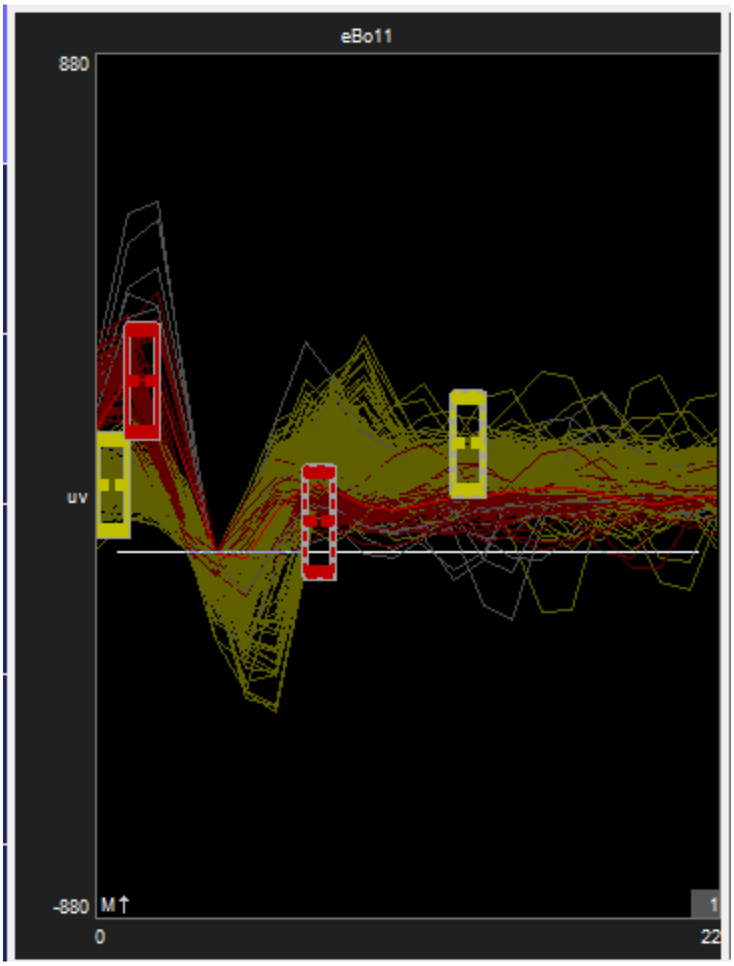
You can also right-click the plot at the desired threshold location and choose **Set Threshold Here** from the menu to move the threshold to that location on one channel. You have the option to apply this new location to all channels in manual thresholding mode.

Right-click the pile plot or threshold display and use the auto/manual threshold options to change the threshold mode of an individual channel.

Box Sorting Using the Pile Plot

Pair of color-coded boxes (one solid and one dotted) are used to classify each unit. In order to be classified as a particular unit, the following is required:

- Candidate waveforms must enter the solid box only one time.
- Candidate waveforms must contain data points that pass through both boxes in the pair.
- One digitized point of the candidate waveform must exist in each box.



Box Sort Waveform Space

To add a box pair:

- Press and hold the **Ctrl** key and double-click to add a new box pair to the pile plot.

A sort code is automatically assigned to the newly added box pair. Click and drag the vertices to adjust the boundaries of the boxes or to move it. To remove a pair of boxes, drag one of the boxes outside of the vertical boundaries of the plot and release.

Important

If a waveform passes through more than one box pair, sort code priority is assigned based on the sort code number. This means that the lower sort code will win in the event that a waveform passes through more than one box pair.

Applying Sorts to New Data

Sort codes are not saved to the data tank until you apply the sorting parameters. You can re-sort or make adjustments as needed to get the best results.

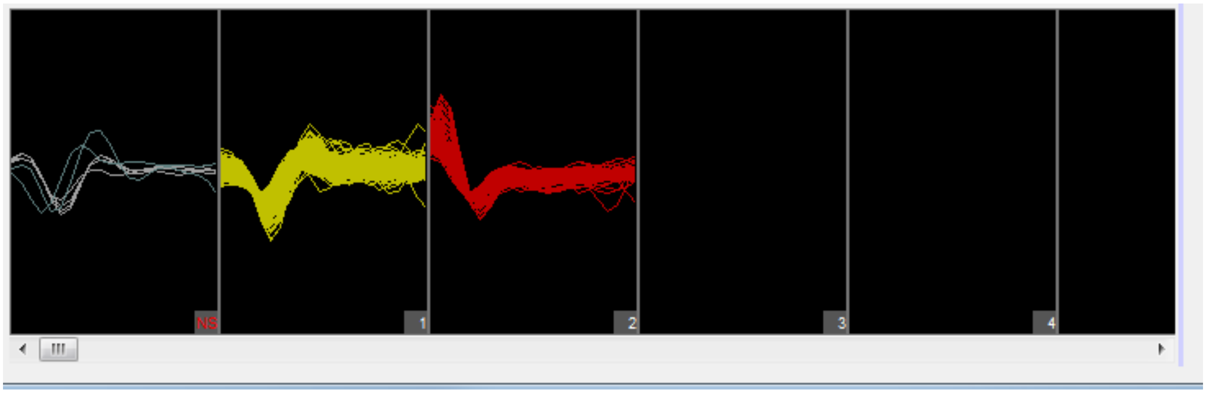
Click the **Hardware Sort** button to send the sorting parameters to the hardware and begin saving sort codes to the tank. Sort codes are applied as new data is acquired. While this button is down, changes in sorting parameters in the display will be applied automatically to new data.

Locking Channels

Click the **Lock All** button to lock the boxes for all channels, or right-click individual channels and choose **Lock**.

Click the **Unlock** button to unlock all channels, or right-click individual channel plots and choose **Unlock**.

The Unit Display



Unit Display

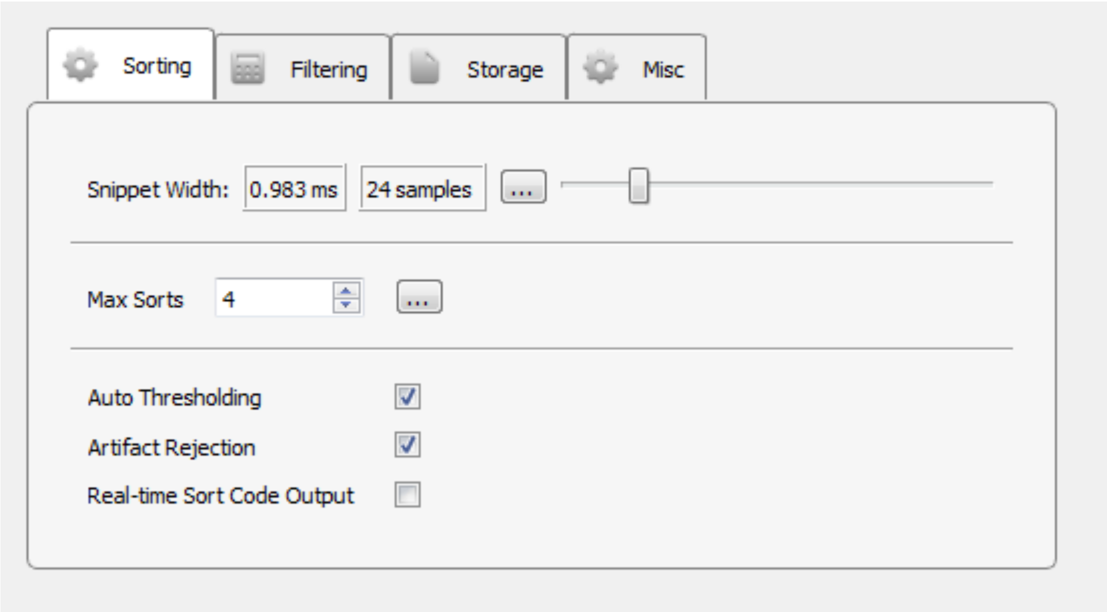
In the unit display, candidate waveforms from the currently selected channel are grouped by sort code. Unsorted (sort code 0) and outlier (sort code 31) waveforms are displayed to the left with the label NS.

The maximum number of sort codes (up to five) that can be sorted on the hardware is determined by the **Max Sorts** configuration setting. Assigned sort codes larger than this value are displayed in red to indicate they are only visible in the software interface. These waveforms will be given a sort code of 31 (outlier) in the data tank.

The unit display can be used to reassign units to different sort codes by clicking-and-dragging the units.

Box Spike Sorting Configuration Options

Sorting Tab



Sorting Options Tab

Snippet Width

Drag slider to select the desired width (displayed in milliseconds and samples) of recorded snippets.

Max Sorts

Events that contain similar shapes are grouped into sorts and given the same sort code. The maximum number of sorts supported in hardware sorting is five. Allowing a larger number of sorts increases processing overhead, but accommodates greater variability in the data set.

Auto Thresholding

In automatic thresholding, the threshold used to record snippets is adjusted in real-time to changes in each channel waveform's RMS. The previous five seconds of data are used in the RMS calculation.

Artifact Rejection

When artifact rejection is enabled, snippets that contain at least one sample greater than the artifact rejection level set on the runtime interface are ignored.

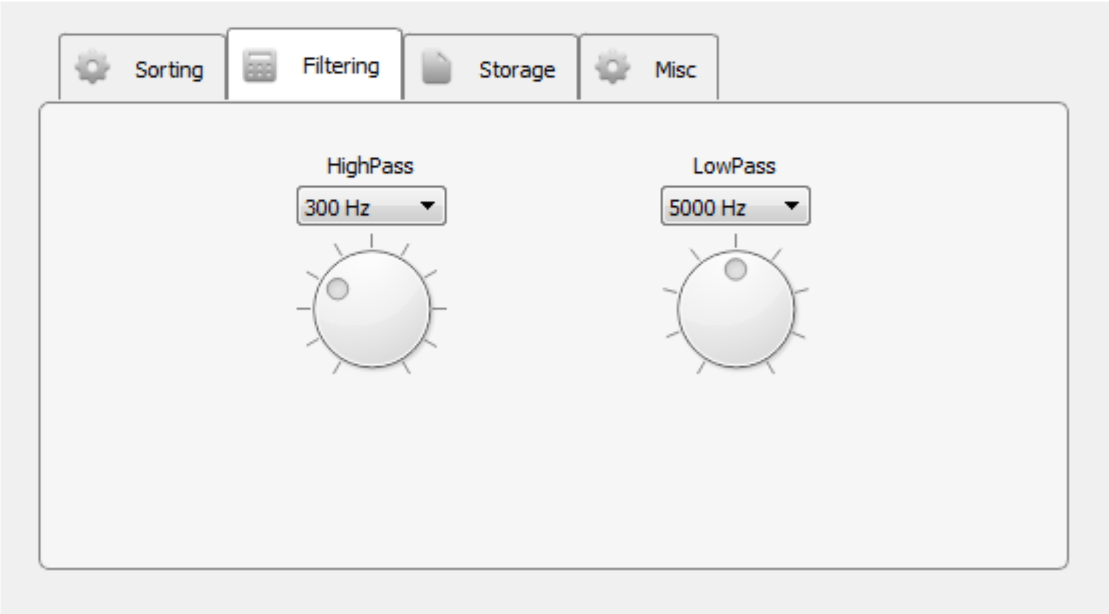
Real-time Sort Code Output

Make the multi-channel integer stream of compressed sort codes available to other gizmos, such as Sort Binner or UDP output.

Note: The sort code output is delayed by (window width + 2) samples from when the threshold is crossed. When artifact rejection is enabled, the sort code output is delayed by an additional window width, so (2 * window width + 2) total samples.

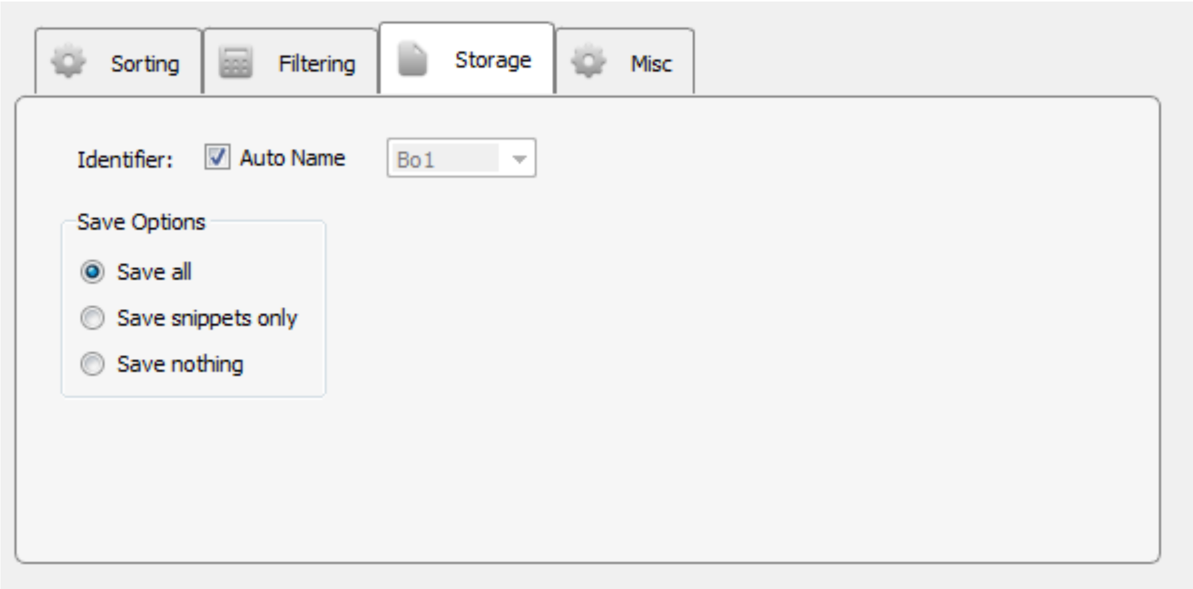
Filtering Tab

The gizmo applies a highpass and lowpass filter to all channels before spike detection. The runtime interface includes controls for dynamic adjustments to the filter settings. You also set default values in the Filtering tab.



Filtering Options Tab

Storage Tab

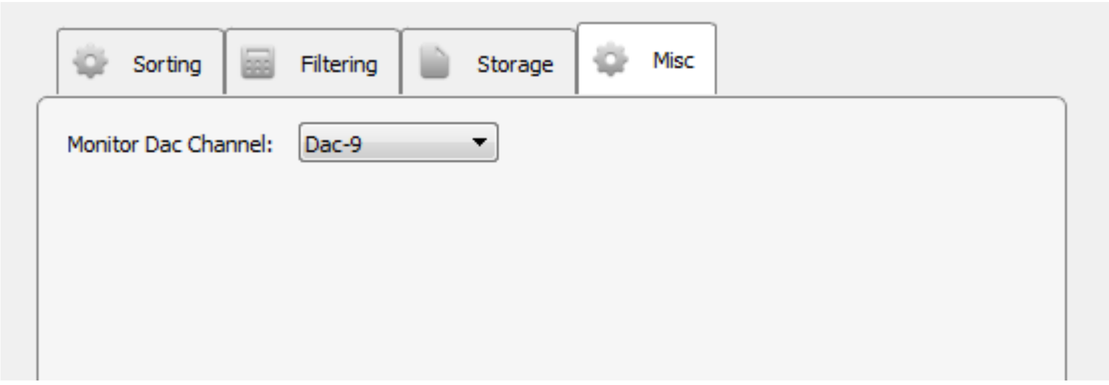


Storage Tab Options

Save Options

Select whether to save only snippet waveforms or to include the plot decimated waveforms used by the sorting gizmo, or to save nothing at all. The waveforms will still be displayed in the runtime interface and data plots but will not be saved to disk.

Misc Tab



Misc Options Tab

Monitor DAC Channel

Select an output channel to send the monitor signal to, or set to **Disable** to turn monitoring off.

Delay

Common Use Cases

Adds a fixed or dynamic delay to any input signal. This gizmo is useful for triggering optogenetic, auditory, or other stimuli a programmed time after an event of interest occurs.

Outputs	
Main	Single or multi-channel delayed signal

Gizmo Help Slides

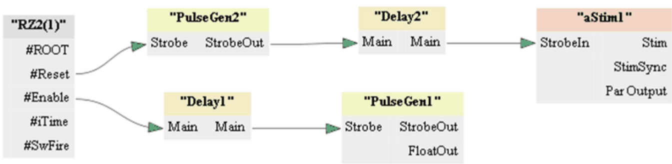
Quick Help Slide 1

Delay — Adds a fixed or dynamic delay to any input signal

Trigger stimulation after baseline recordings

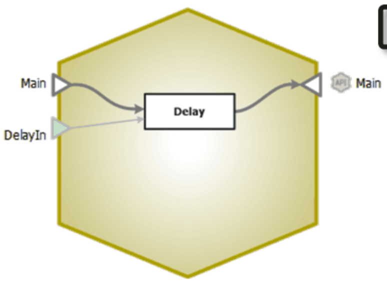
Create variable stimulation delay paradigms for behavioral trials

Two paradigms. Top: Trigger a pulse, then an audio stimulus at a fixed delay after the pulse onset. Bottom: Trigger a pulse train at a set delay after the onset of recording



Main: Signal to delay, can be any type

DelayIn: Optional floating point value from another gizmo to specify the delay time



Main: The delayed signal

Quick Help Slide 2

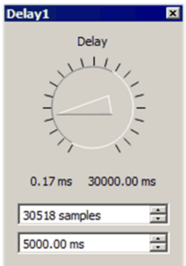
Delay

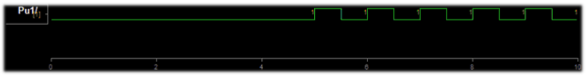
User Interface, API, Data Stored

Dynamic control of signal delay. Must be set before Delay gizmo receives input signal and delay is initiated.

Delay — Value of input delay in milliseconds (dynamic delay only) {Read/Write}

None





Example: PulseGen train trigger delayed 5 seconds from onset of recording.

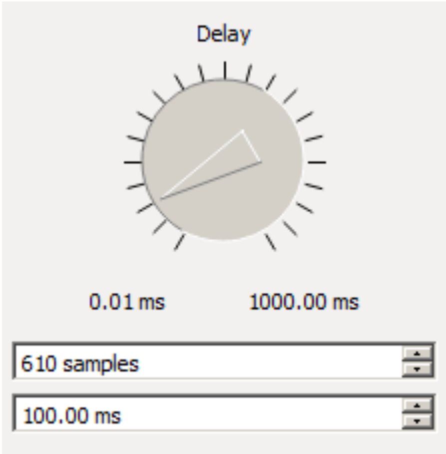
Reference

The Delay gizmo takes any single or multi-channel input and adds a fixed or dynamic delay to the signal.

The Runtime Interface

Delay Tab

The runtime interface is available when Delay Mode is Dynamic and Control Source is Widget. The value of the signal delay is then adjustable by the user at runtime and is also available through the SynapseAPI.

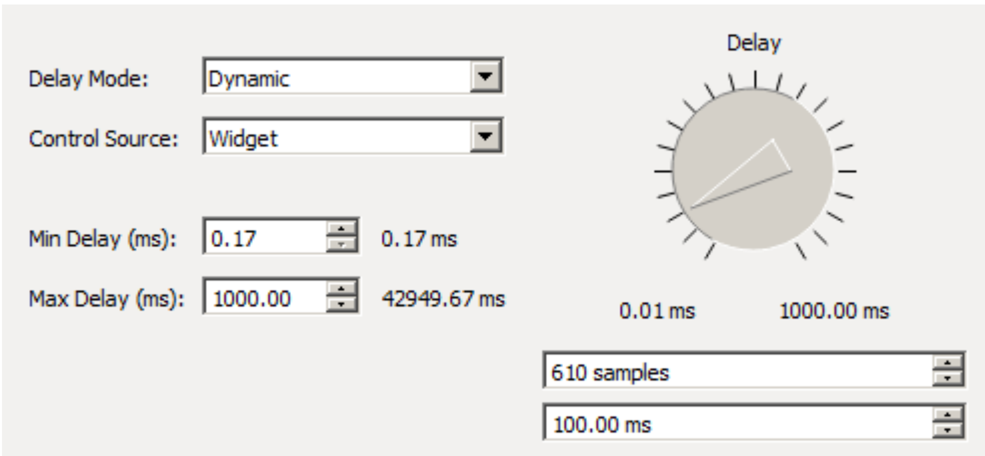


Delay Runtime UI

In the above example, the min and max delay values were set to 0.01 ms and 1000 ms at designtime and the current delay value is 100 ms. Because the device sampling rate was ~6K, the actual number of real-time samples shows 610.

Delay Configuration Options

Options Area



Delay Configuration Options

The Delay gizmo can take either a single channel or a multi-channel input. The signal type of the input determines the signal type of the output.

In **Static Mode**, the delay value set at designtime is the signal delay used at runtime. In **Dynamic Mode**, the delay value can either be controlled by a Widget at runtime or by a Gizmo Input by setting the Control Source.

The min and max delay bounds are set at designtime. In **Widget** mode, the bounds set the min and max values of the knob. Also, use **Widget** mode if controlling the delay value via the SynapseAPI. The absolute maximum delay available in any mode is displayed to the right of the Max Delay spin box. This is dependent on the type of input signal and the device sampling rate.

Electrical Stim Driver

Common Use Cases

Create up to four stimulation voices for single-ended or bipolar stimulations outputs on a target device, such as an IZ2 or IZV. Create monophasic or biphasic waveforms with charge balancing options. Use this gizmo for design of interesting electrical stimulation waveforms.

Data Stored	
Epoc (optional)	Parameter values when triggered
Stream (optional)	Raw stimulus waveform
Outputs	
Stim	Stimulus waveform, single channel floating point
StimSync	Logic signal when electrical stimulation is active
VoiceAct	Integer code of the active voice(s)
PulseAct	Integer code of the active voice pulses
Par Output	Full parameter stream

Gizmo Help Slides

Quick Help Slide 1

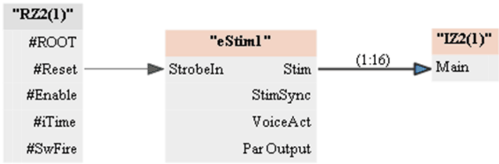
Electrical Stim Driver — Create up to four stimulation voices for single-ended or bipolar stimulations outputs on a target device, such as an IZ2 or IZV. Create monophasic or biphasic waveforms with charge balancing options

Neural stimulation experiments

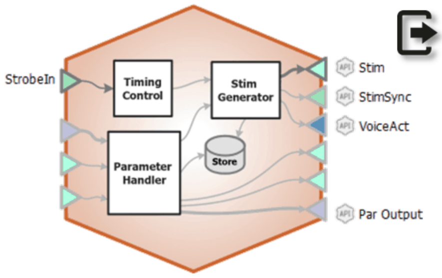
Response-reward behavioral paradigms

Complex stimulation paradigms

RZ2 processor #Reset tied to eStim1 gizmo for manual control of stimulus presentation. Signal sent to IZ2 stimulator to present the stimulus on one of 16 channels.



StrobeIn: Optional input to control presentation timing



Stim: Stimulation signal(s). May also contain channel information for IZV

StimSync: Logic signal that is high during a stimulus presentation

VoiceAct: Integer bit mask indicating which voice(s) are currently active

ParOut: Parameter values for the given presentation

Quick Help Slide 2

Electrical Stim Driver

User Interface, API, Data Stored

Manually strobe stimuli presentations. Change waveform parameters during runtime

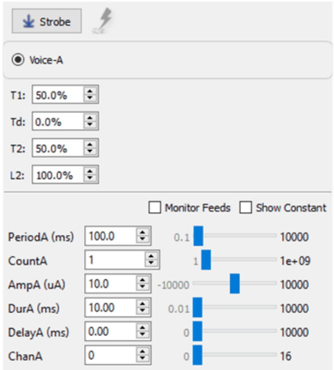
Strobe — In Manual mode, strobe the gizmo to give a stimulus presentation {Write}

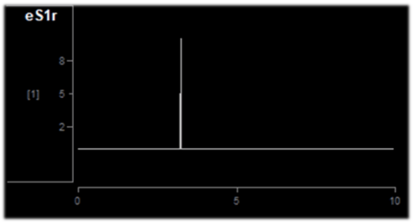
StimActive — Logic high during a stimulus presentation {Read}

Mute — If enabled, mute all voices {Read/Write}

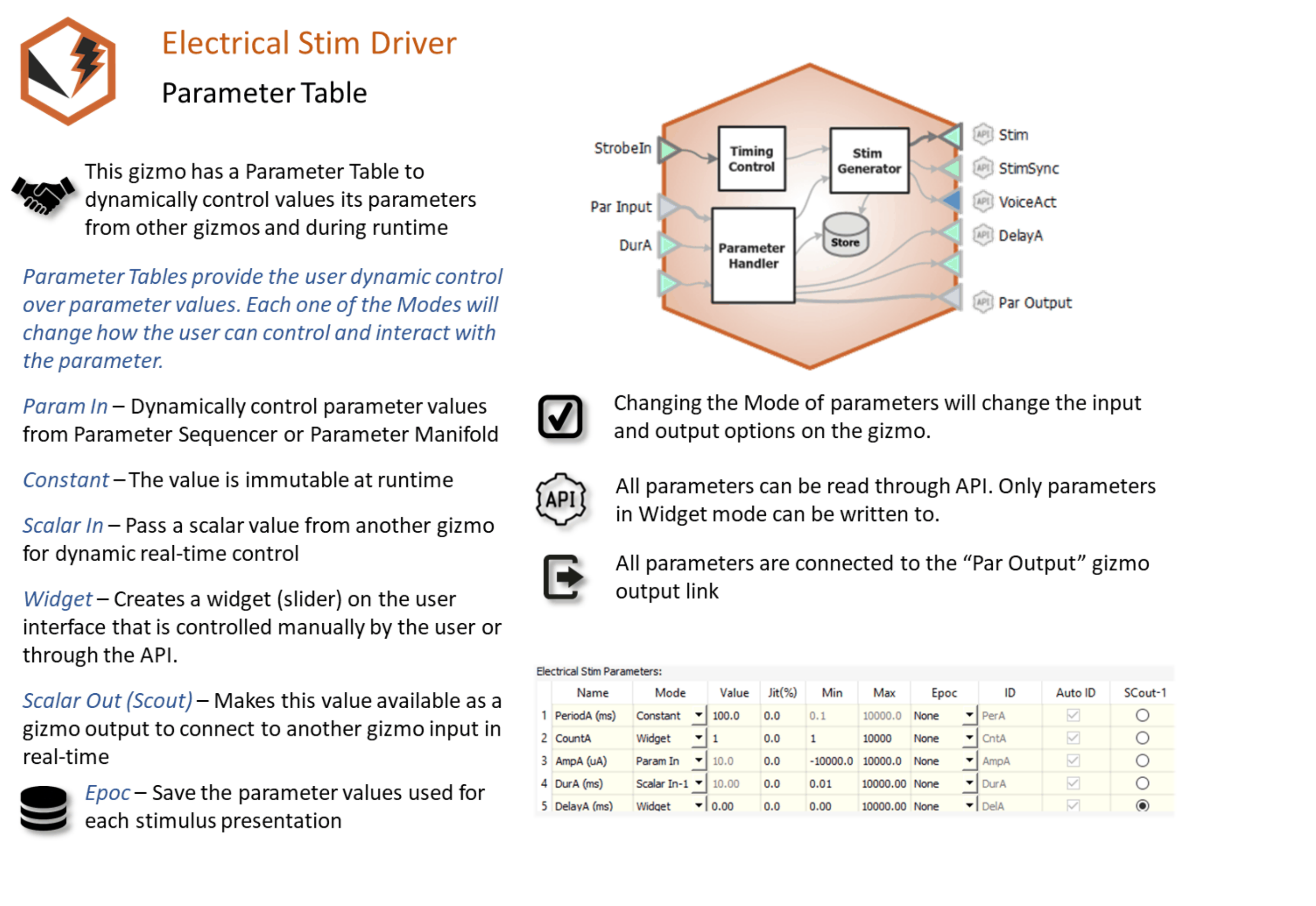
eS1r — Continuous waveform containing stim voices {Stream}

eS1p — List of all stimulus parameters stored when stim occurs (optional) {Strobe}





Quick Help Slide 3



	Name	Mode	Value	Jit(%)	Min	Max	Epoc	ID	Auto ID	SCout-1
1	PeriodA (ms)	Constant	100.0	0.0	0.1	10000.0	None	PerA	☑	○
2	CountA	Widget	1	0.0	1	10000	None	CntA	☑	○
3	AmpA (uA)	Param In	10.0	0.0	-10000.0	10000.0	None	AmpA	☑	○
4	DurA (ms)	Scalar In-1	10.00	0.0	0.01	10000.00	None	DurA	☑	○
5	DelayA (ms)	Widget	0.00	0.0	0.00	10000.00	None	DelA	☑	◉

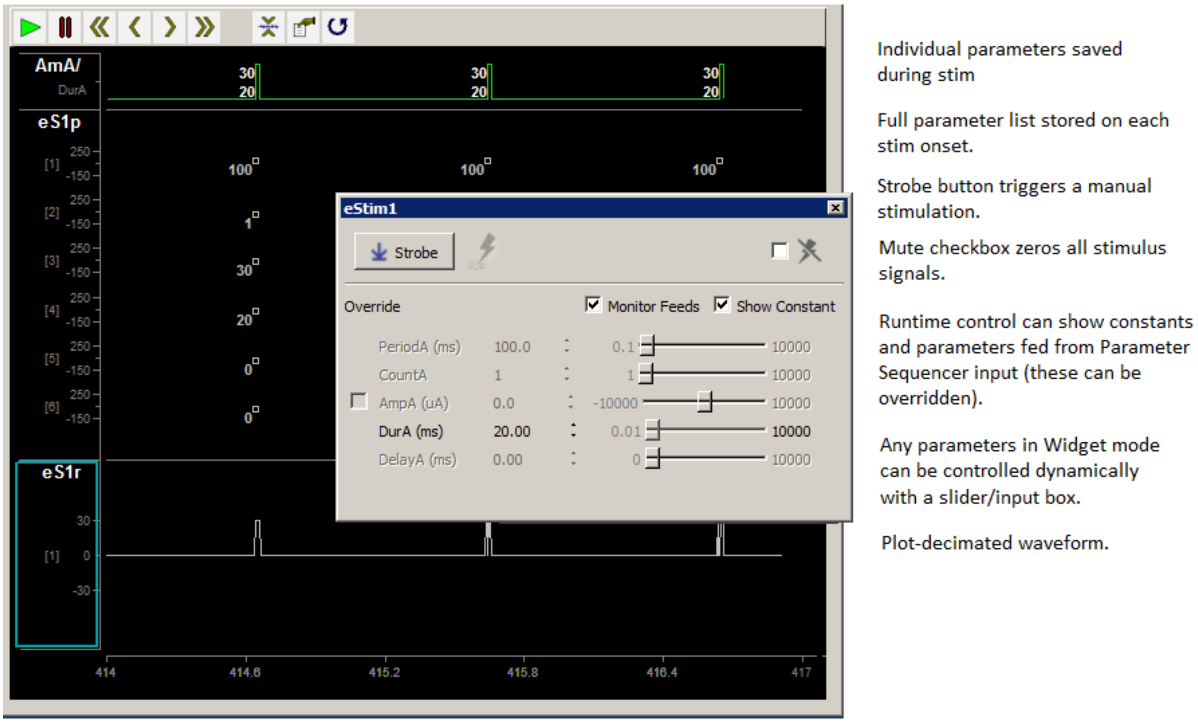
Reference

The Electrical Stim Driver gizmo configures timing, parameter handling, and electrical stimulation generation with up to four independent stim patterns (voices). It allows dynamic control of stim timings, amplitudes, delays relative to trigger onset, and presentation channels. Output can directly control external IZ2 or IZV (SIM) stimulator device.

> **Important**
>
> For electrical stimulation design with the IZ2, this gizmo replaces the Electrical Stimulation gizmo and Injector gizmo.

Electrical stimulation waveforms are comprised of square waves that can vary in duration, level, phase, delay, and stim channel. The gizmo provides static or runtime control of stimulus parameters and options to store individual parameters, the parameter list, and raw waveform. Timing pulse can also be output for secondary control or storage.

Electrical Stim Driver Runtime Interface

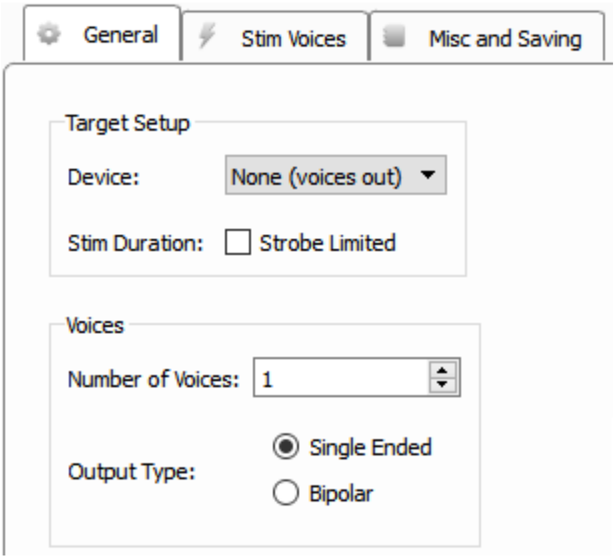


Runtime Tabs with Various Stimulus Stores Enabled for Illustration

The illustration above shows the different ways stimulus information can be stored with the Electrical Stim Driver gizmo. Whichever stores you chose to include will be added to the runtime plot alongside the recording plots. If enabled in the gizmo configuration, a control tab is added at runtime.

Electrical Stim Driver Configuration Options

General Tab



General Tab (Target Device is None)

Target Device

When set to **None (voices out)**, the Stim output link from the gizmo will contain the voices only.

If **Stim Duration** is set to Strobe Limited, then the StrobeIn input to the gizmo controls the number of presentations. When triggered, stim presentations continue until the Count* parameter is reached or the StrobeIn input goes low, whichever happens first.

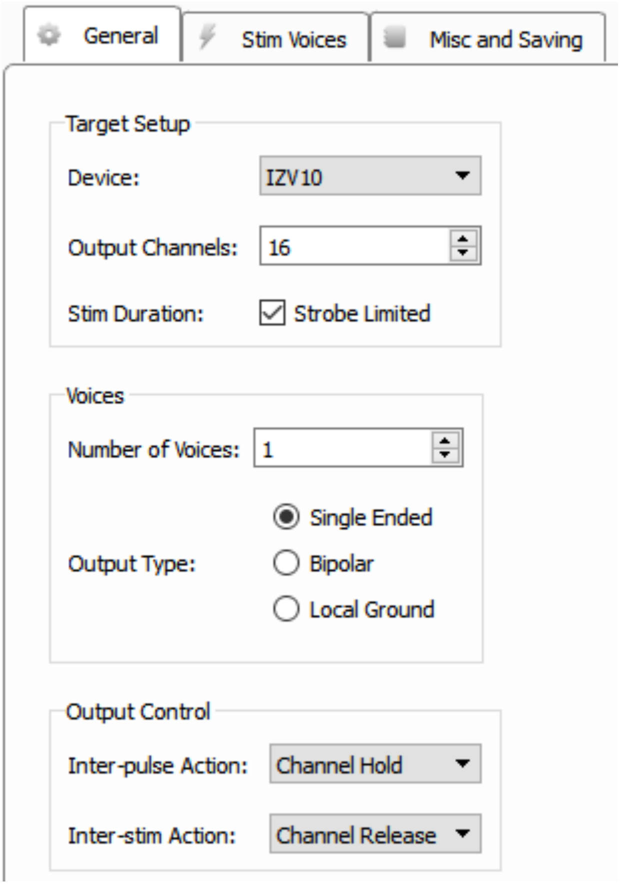
Bipolar mode adds inverted versions of the configured voices to the output signal.

The Electrical Stim Driver will typically be used to control an IZ2/IZ2M or IZV (SIM) stimulator.

When set to **IZ2** or **IZV10**, choose the total number of stim channels on your IZ.

Choose the number of unique stim waveforms (**Voices**) to present (up to 4). If **Bipolar** is enabled, you can have up to 2 independent patterns and their inverted waveform on the paired channel.

IZV (SIM) Configuration Options



Local Ground mode creates a paired channel for each voice that is shorted to the sub-amp ground to make that paired channel the return path.

You can control the behavior of the IZV channels between individual pulses in a train ('Inter-pulse Action') and between each burst ('Inter-stim Action').

Channel Hold keeps the stim voice connected to the output channel and set to zero

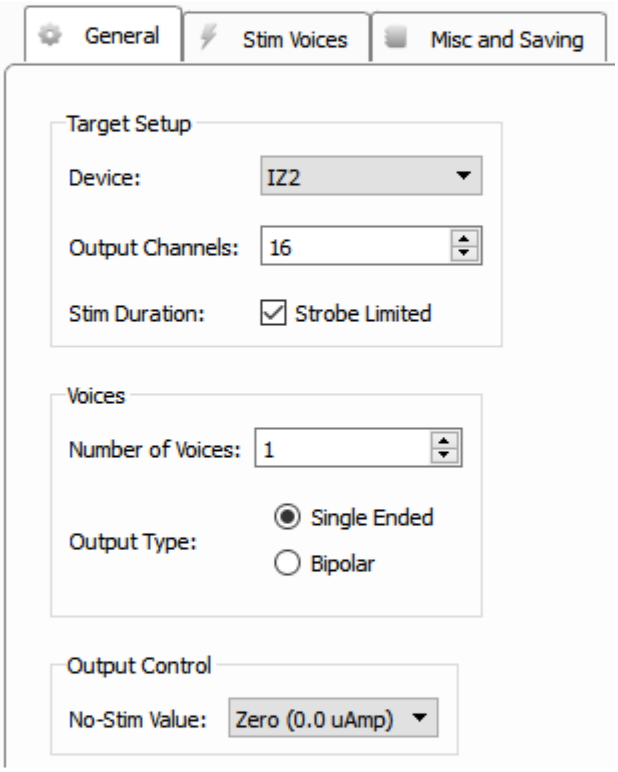
Channel Release open circuits the channel.

Discharge activates a ground clamp (10 kOhm resistor to ground) passive discharge on the stim voice between presentations.

Ground shorts the channel to ground between presentations.

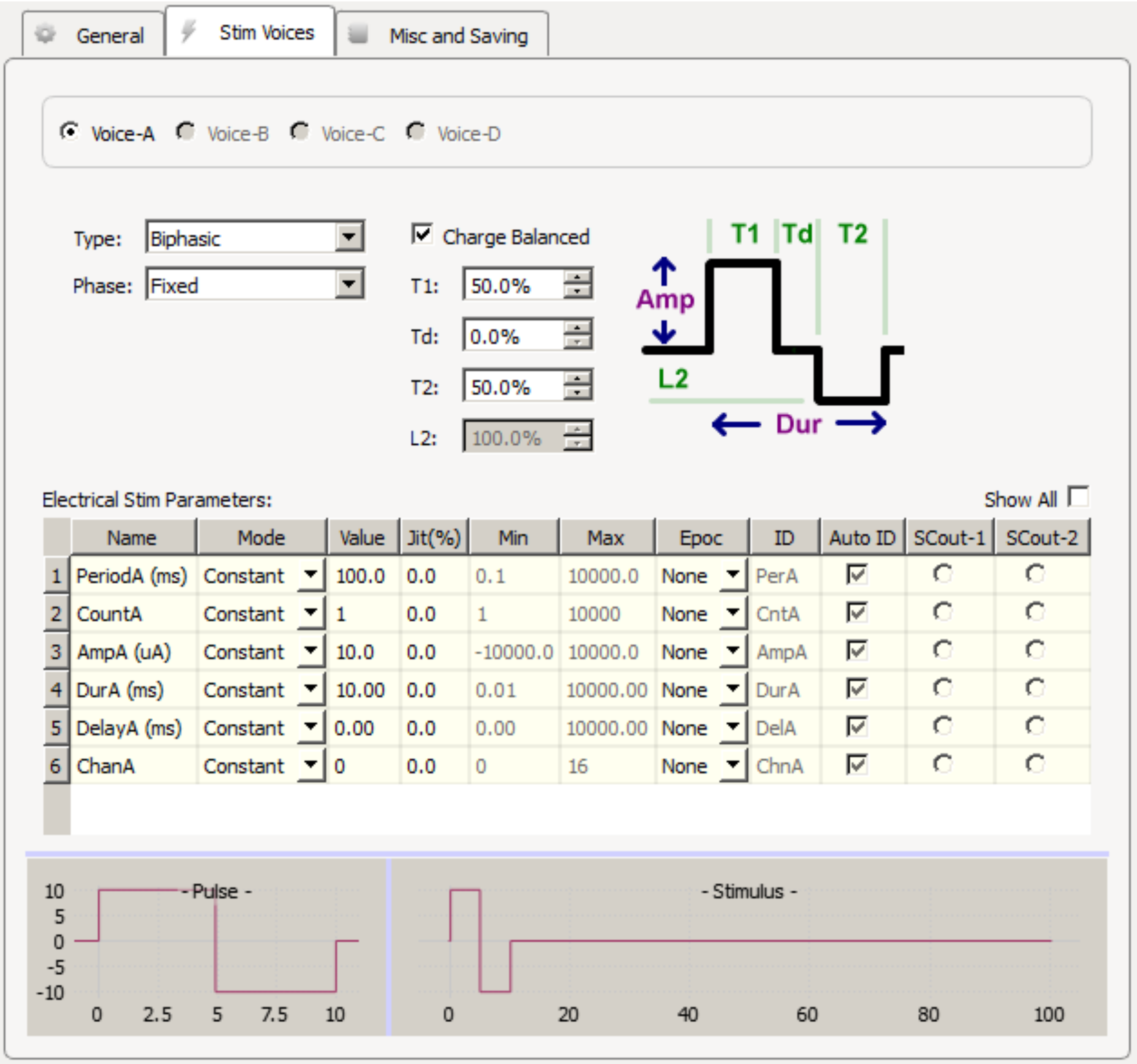
By default, the IZV channels will be open circuit (**Channel Release**) in between bursts and set to zero current in between pulses in each burst (**Channel Hold**).

IZ2/IZ2M Configuration Options



The **No-Stim Value** option tells the IZ2 channels that aren't actively stimulating how to behave. **Zero (0.0 uAmp)** sets the control signal to 0 but keeps the channel connected to the stimulator. **Open Relay** open circuits the channel completely (recommended). **Ground** shorts the control channel to ground (Note: the **Ground** option is not compatible with IZ2M or IZ2MH devices).

Stim Voices Tab



Stim Voices Tab

Design the waveform shape of each voice. For Monophasic waveforms, the amplitude and duration are controlled by the Amp and Dur parameters.

Biphasic waveform parameters are based on percentage calculations of the Amp and Dur parameters. T1 and T2 are the durations of the phases, with Td the time in between the phases. L2 is the inverted level of the second phase as a percentage of Amp. Check the **Charge Balanced** box to automatically compute L2 so the total current delivered is zero. This minimizes tissue damage and electrode corrosion for electrical stimulation applications.

Phase

By default, the amplitude of the pulse defines the phase of the stimulus. The **Phase** setting can control whether to keep the phase Fixed, or to apply an alternating phase per Pulse (toggle between 1 and -1 scale factor on each pulse in the train), or per Stim (toggle the scale factor once before the stimulus train begins).

If the Output Type is **Bipolar** or **Local Ground**, a second channel appears for each voice to control the paired channel number.

Electrical Stim Parameters

These define the waveform shapes for each boice.

> **Tip**
>
> See Using Parameters for more information on controlling parameter tables.

Mode

In the Mode column, you can choose to make an individual parameter Constant, controlled by a runtime Widget, or controlled by a Parameter Input (**Param In**) or one of two possible Scalar Input lines (Scalar In-1, Scalar In-2).

Value Columns

Enter values in the Value, Jit% (Jitter), Min, and Max columns to set the constant value or to set the initial value when a widget control will be used.

Epoc

In the Epoc column, you can choose to save the individual parameter value on stimulus onset/ offset or on individual pulse onset/offset.

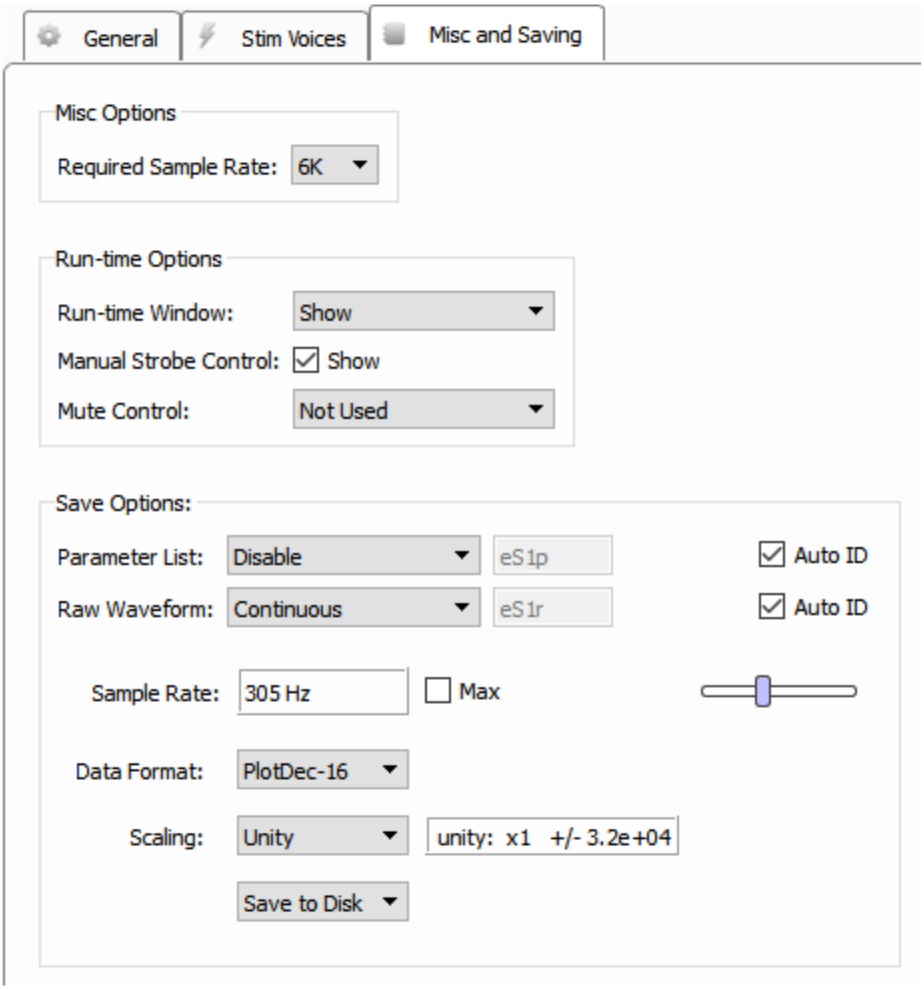
ID and Auto ID check box

Synapse automatically generates a store name. TDT recommends using Auto ID to ensure no store names are duplicated. To make your own store names, clear the **Auto ID** check box.

SCout-1 and SCout-2

Select the radio button in the desired row to feed the parameter to an output line.

Misc and Saving Tab



Misc and Saving Tab

Required Sample Rate

The minimum rate required. Synapse looks through the entire experiment and your rig and sets the sample rate according to this and other limiting factors.

Run-time Window

By default a runtime tab is added in preview or record mode. The contents of the tab are defined with configuration options on the General and Parameter options tab. Choose 'Hide' to completely hide this tab. When a stim voice is set to Biphasic, you can also control the T1, Td, T2 parameters at runtime. To hide those controls but show the other controls in the parameter table, choose 'Hide Biphasic Controls'.

Manual Strobe Control check box

When selected a manual strobe control is added to the runtime eStim tab.

Mute Control

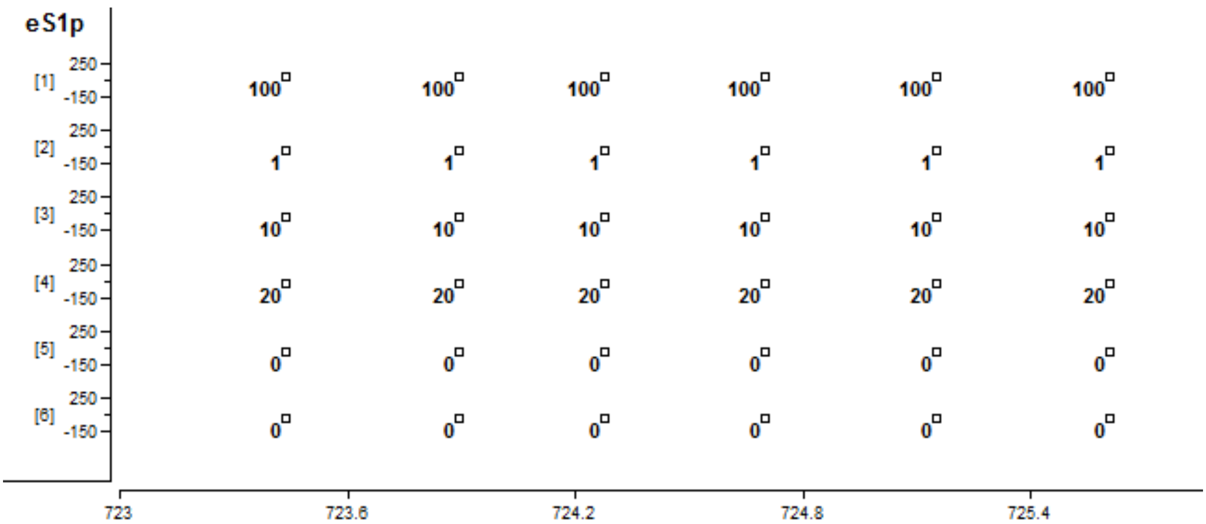
Select the default behavior of the runtime mute control. Mute allows you to temporarily zero all stimulus output during runtime. You can choose to hide or show the control and, if show, set the default start state. For controlling an IZ2M/IZ2MH or IZV cards that require arming before you can stimulate, you'll likely want to set this option to Default Muted so that no stimulation comes out when the experiment begins.

Save Options

The options in this area configure stores that can be generated natively within the gizmo.

Parameter List

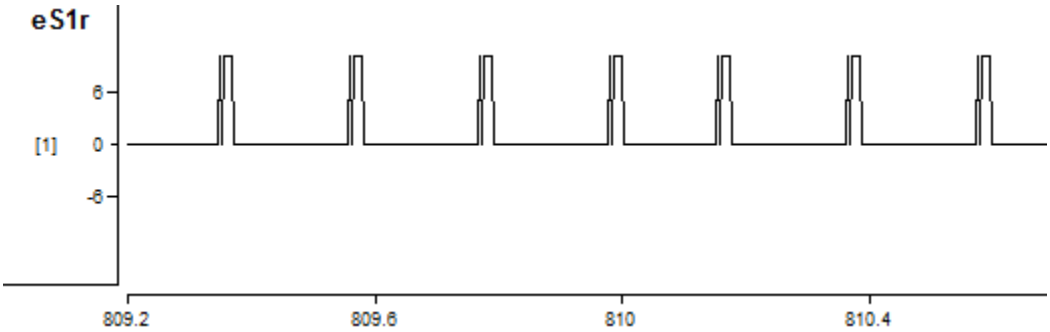
Select whether to store the value of all parameters, at each stimulus onset. This generates a multi-channel list of scalar values. The channels map directly to the rows of the parameters table on the Stim Voices tab. By default, some parameters are hidden in the table, but values are stored for all parameters.



Parameter List Store in the Runtime Plot

Raw Waveform

Select whether to store a copy of the raw stimulus waveform(s). You can choose to store continuously or only when the stimulus is active. By default the Data Format is set to plot-decimated, which is highly decimated for viewing on your computer monitor. It takes short chunks of points, finds the max and min values, and only keeps those. You can see this effect in the image below. The actual output is a square wave, but during the rising edge it plots the max and min for that chunk of time in the plot decimated view.



Raw Stimulus Waveform Store in the Runtime Plot

This is fine for monitoring the peak signal output and requires very low bandwidth. The plot will not contain all of the signal information but will capture the maximum/minimum signal amplitudes. Change the Data Format to Float-32 and increase the Sample Rate if you need a better resolution view.

VoiceAct Output Link

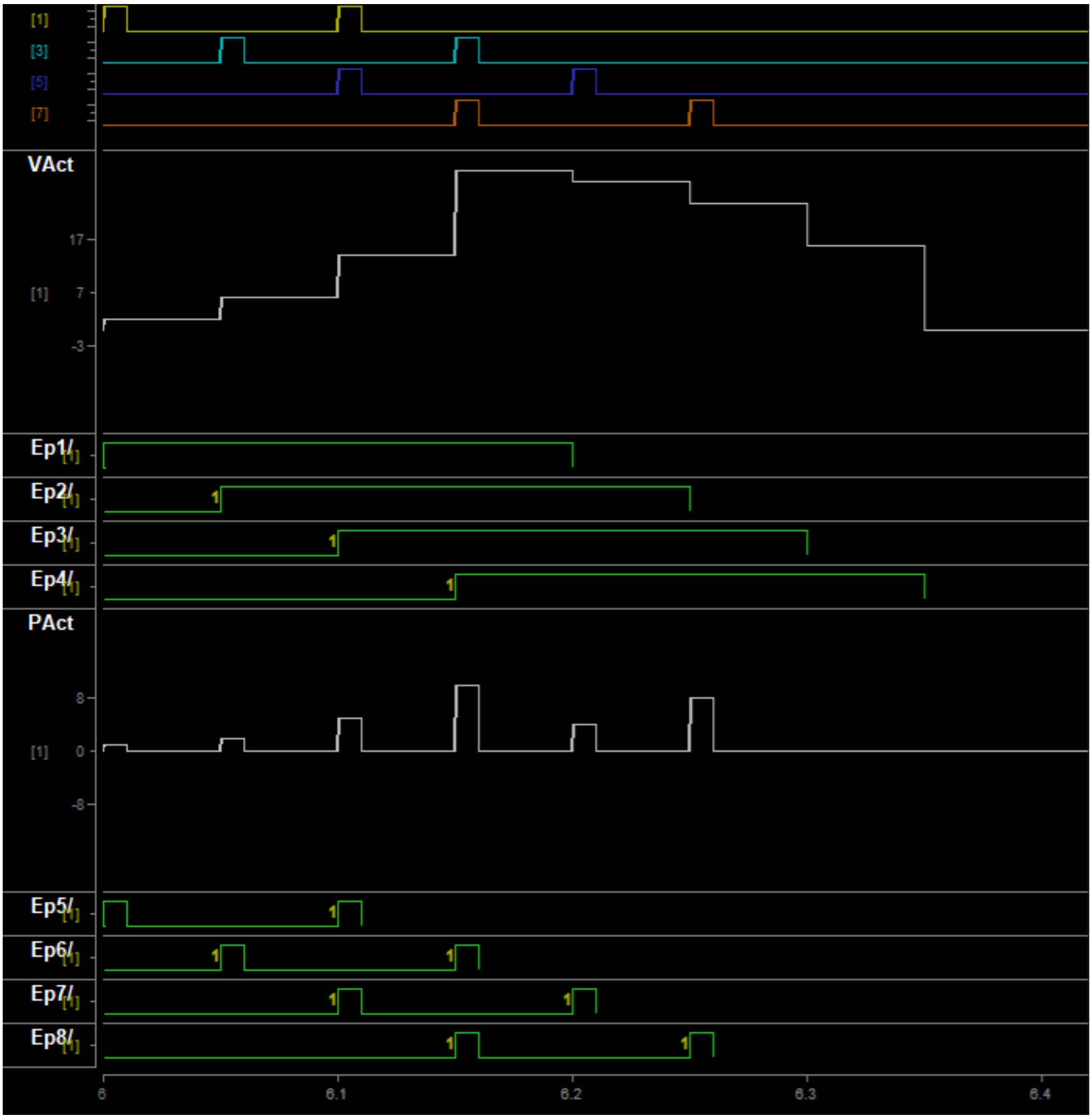
The VoiceAct output is a bit mask where the first bit is the stim trigger, and the next four bits represent whether voices A, B, C, or D respectively are currently stimulating.

PulseAct Output Link

The PulseAct output is a bit mask where the first bit is the stim trigger, and the next four bits represent whether voices A, B, C, or D respectively are sending a non-zero waveform.

In the image below, the four voice output stimulation waveforms are at the top.

- VAct is the VoiceAct output
- Ep1-Ep4 are epocs saved "On Stim" for each voice
- PAct is the PulseAct output
- Ep5-Ep8 are epocs saved "On Pulse" for each voice



Comparison of VoiceAct and PulseAct output links

Electrical Stimulation

The Electrical Stimulation gizmo configures timing, parameter handling, and electrical stimulation generation.

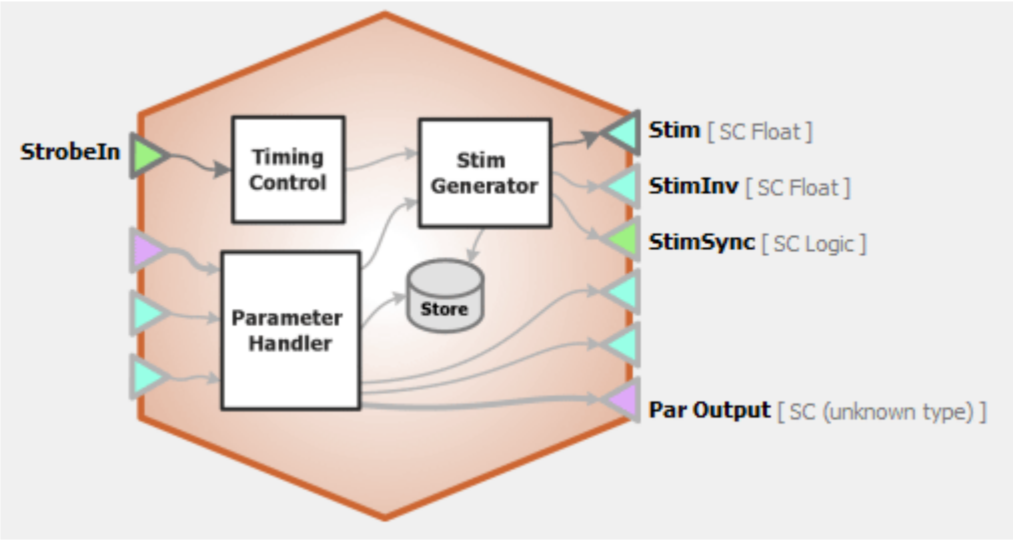
Important

The Electrical Stimulation gizmo was replaced by the Electrical Stim Driver gizmo. It is only available if you enable Deprecated gizmo in Menu → Preferences.

Data Stored	
Epoc (optional)	Parameter values when triggered
Stream (optional)	Raw stimulus waveform
Outputs	
Stim	Stimulus waveform, single channel floating point
StimInv	Inverted stimulus waveform
StimSync	Logic signal when electrical stimulation is active
Par Output	Full parameter stream

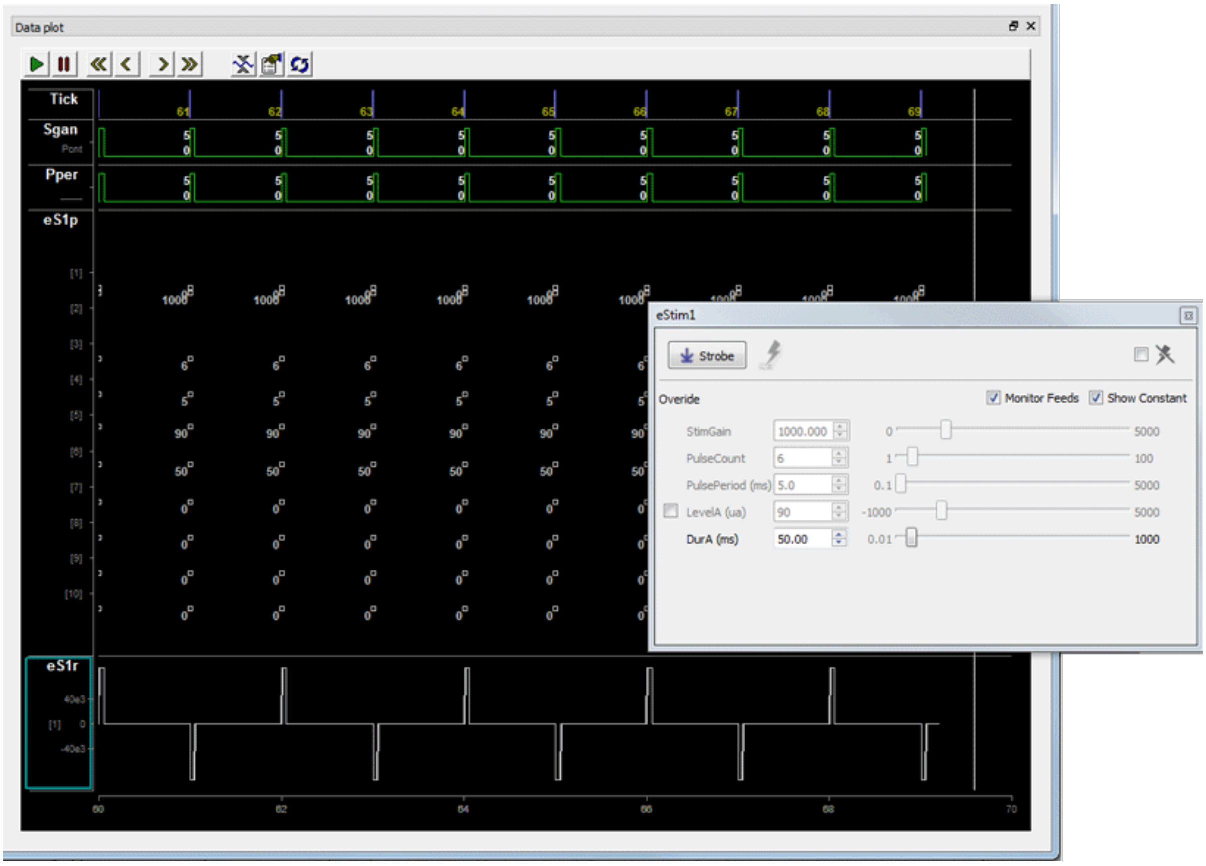
Reference

Electrical stimulation waveforms are comprised of square waves that can vary in duration, level, and phase. The overall stimulation duration can be set by a fixed duration, based on a strobe or based on pulse count. The gizmo provides static or runtime control of stimulus parameters and options to store individual parameters, the parameter list, and raw waveform. Timing pulse can also be output for secondary control or storage. Both the stimulus and the inverse of the stimulus are gizmo outputs.



Electrical Stimulation Block Diagram

Electrical Stimulation Runtime Interface

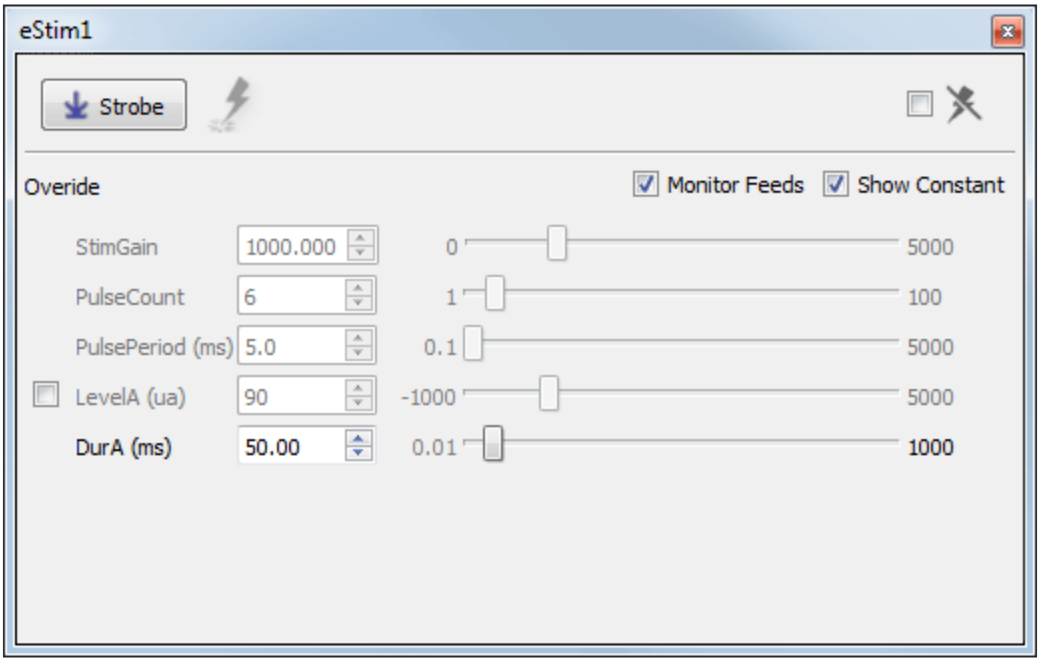


Runtime Tabs with Various Stimulus Stores Enabled for Illustration

The illustration above shows the different ways stimulus information can be stored with the Electrical Stimulation gizmo. Whichever stores you chose to include will be added to the runtime plot alongside the recording plots.

If enabled in the gizmo configuration, a control tab is added at runtime. Parameters that can be controlled dynamically are shown in black (active). You can enter a value in the field, use up and down arrows, or drag a slider to modify to parameter value.

The illustrations above and below, show the tab floated and with all the options shown. You can show only the elements you need or hide the entire control.



eStim Runtime Tab - Floated

Click and release the **Strobe Button** to trigger a manual strobe pulse.

Select the **Mute Button** check box to zero the stimulus signal.

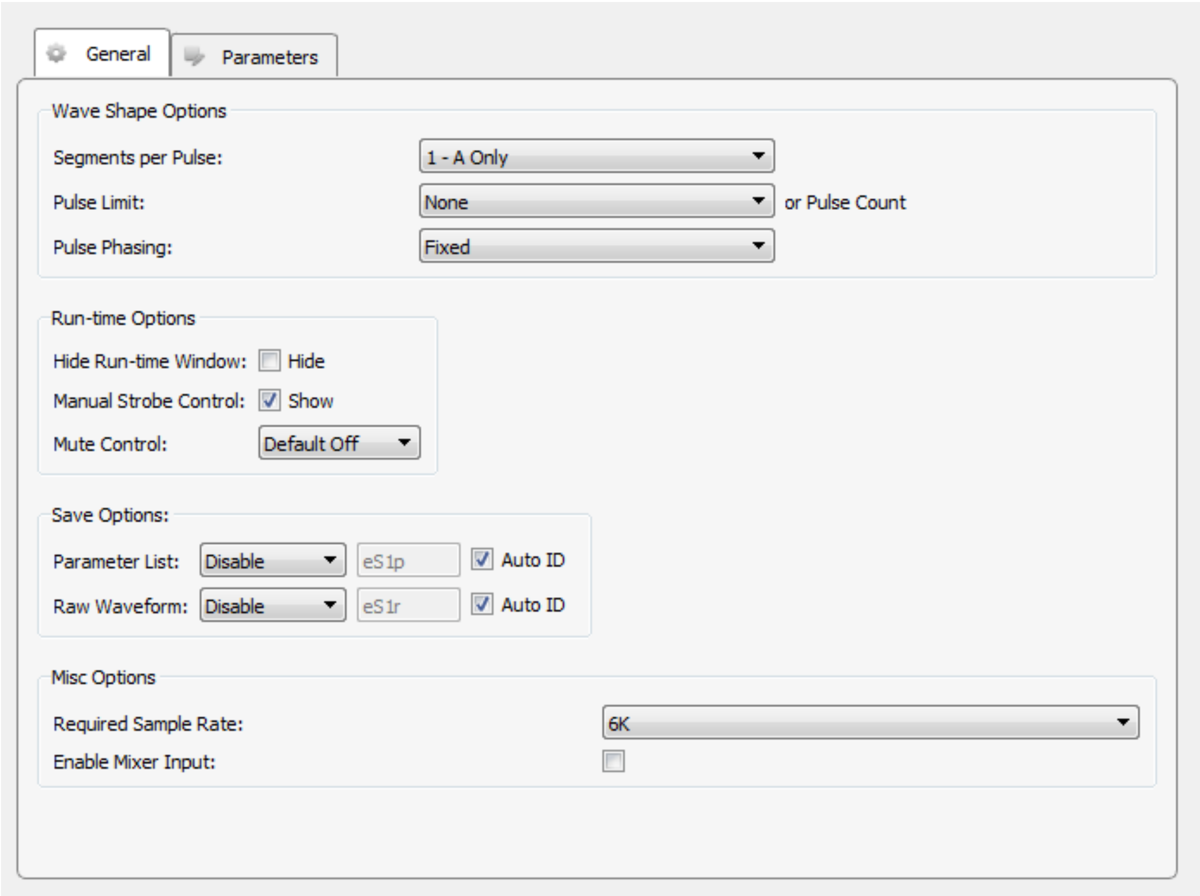
Select the **Monitor Feeds** check box to show stimulus parameters controlled by an input signal. Also adds an Override column and check box to the left. Select the **Override** check box to adjust the parameter value manually instead of using the input signal.

Select the **Show Constant** check box to display values for parameters set to Constant. They will appear gray.

Electrical Stimulation Configuration Options

Use the options tabs to enable/disable optional features and set parameters that will be used to configure the gizmo operation and interface. Changes are not applied until you commit all settings. See The Options Area and Templates for more information on the gizmo name, source, global options, and displaying the block diagram.

General Tab

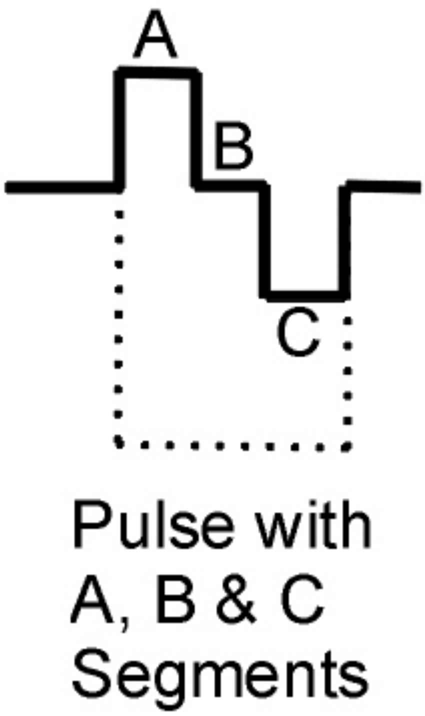


General Tab

Wave Shape Options

Segments per Pulse

Choose the number of segments that make up each pulse. Each pulse can have up to three segments, designated A, B, or C. Level and duration for each segment are configured on the Parameters tab. Examples below illustrate how segments can be used to build various waveform shapes.



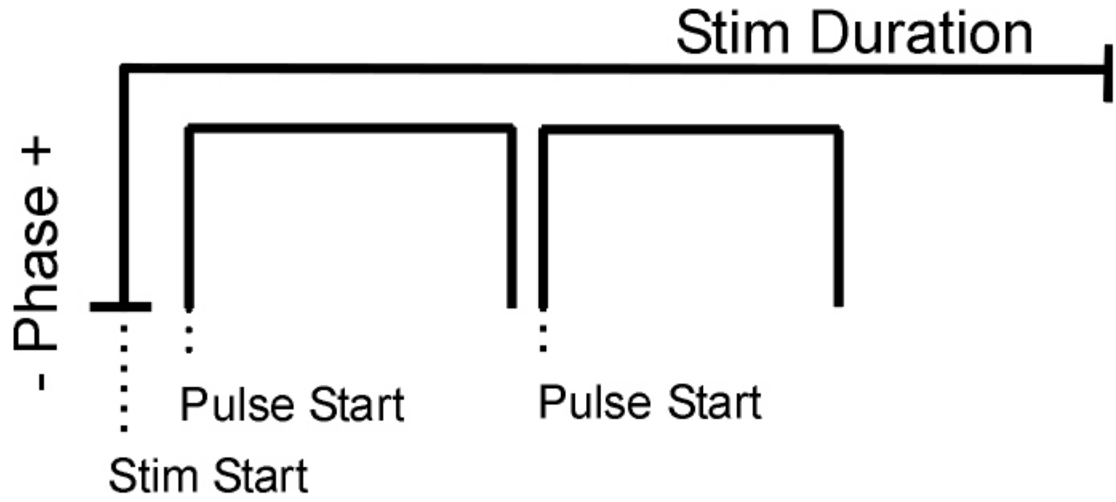
Pulse Examples

Pulse Limit (or Pulse Count)

Select how the stimulus comes to an end, that is, pulses stop. If none is selected, any pulse count value or method will be applied.

Pulse Phasing

By default, the level value of the pulse defines the phase of the stimulus. Pulse phasing can apply an alternating phase (* -1)by pulse or stimulus. If used, it will be applied at the start of a stimulation presentation or the start of each pulse.



Stimulus/Pulse Phase Diagram

Hide Run-Time Windows check box

By default a runtime tab is added in preview or record mode. The contents of the tab are defined with configuration options on the General and Parameter options tab. Select the check box to hide the runtime tab.

Manual Strobe Control check box

When selected a manual strobe control is added to the runtime eStim tab.

Mute Control

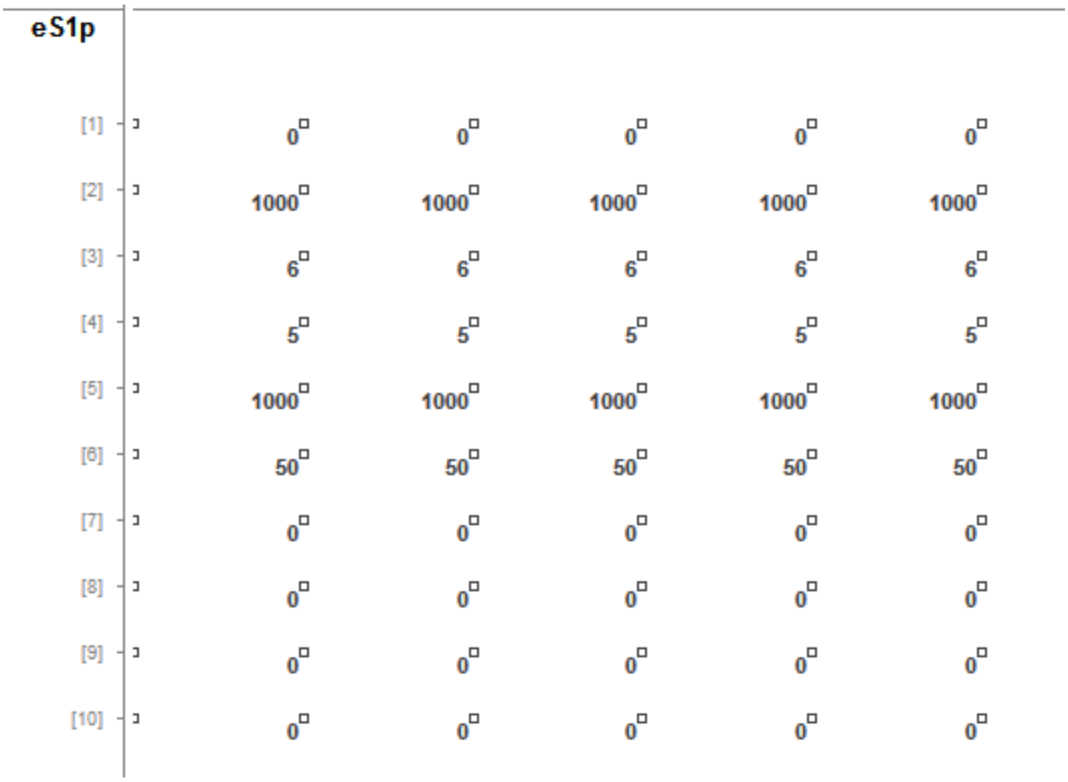
Select the default behavior of the runtime mute control. Mute allows you to mute or temporarily zero the stimulus during runtime. You can choose to hide or show the control and, if show, set the default start state.

Parameter List

Select whether to store the value of all parameters, at each stimulus or pulse onset. This generates a multi-channel list of scalar values. The channels map directly to the rows of the parameters table on the Parameters tab. By default, some parameters are hidden in the table, but values are stored for all parameters.

Auto ID field and check box

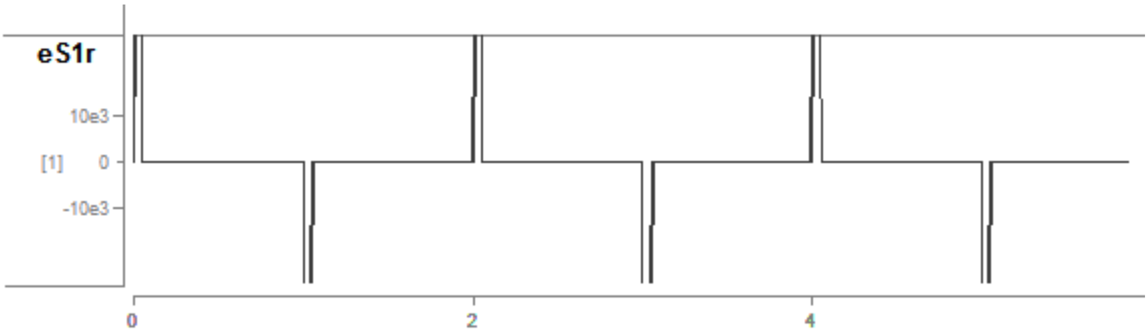
A store name is generated automatically. A "/" is appended to the name to indicate when the full epoc is stored (and is not appended when only saving the onset). To use your own store name, clear the **Auto ID** check box.



Parameter List Store in the Runtime Plot

Raw Waveform

Select whether to store a copy of the raw stimulus waveform. You can choose to store continuously or only when the stimulus is active.

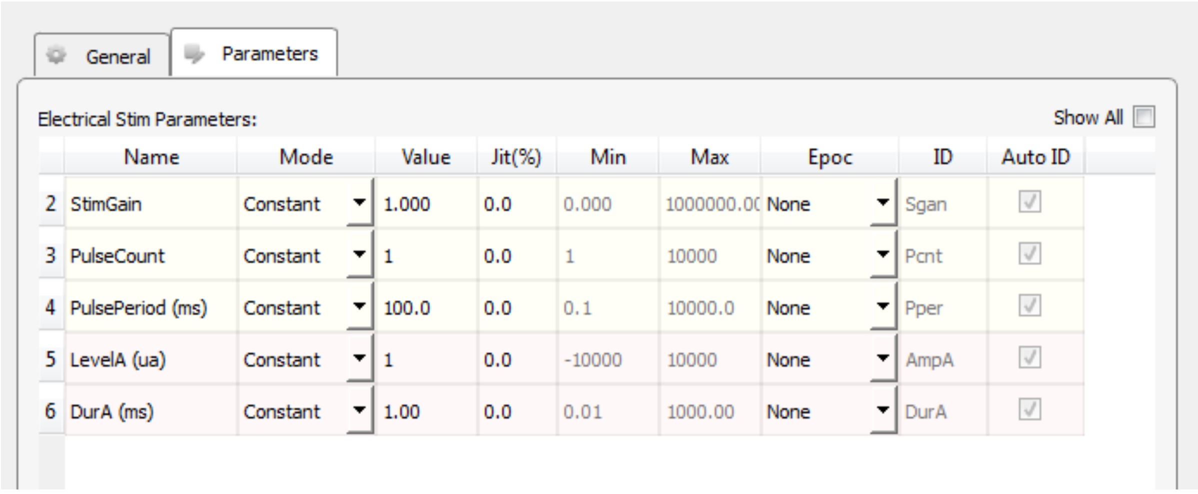


Raw Stimulus Waveform Store in the Runtime Plot

Required Sample Rate

The minimum rate required. Synapse looks through the entire experiment and your rig and sets the sample rate according to this and other limiting factors.

Parameters Tab



Parameters Tab

Electrical Stim Parameters

The table lists parameters relevant to configuring a stimulus. Each row represents a parameter and rows are shown or hidden in response to selections you make on the General tab. Use the **Show All** check box to display hidden rows.

> **Tip**
>
> See Using Parameters for more information on using the parameters table.

Mode

In the Mode column, you can choose to make an individual parameter Constant, controlled by a runtime Widget, or controlled by a Parameter Input (**Param In**) or one of two possible Scalar Input lines (Scalar In-1, Scalar In-2).

Value Columns

Enter values in the Value, Jit% (Jitter), Min, and Max columns to set the constant value or to set the initial value when a widget control will be used.

Epoc

In the Epoc column, you can choose to save the individual parameter value on stimulus or pulse onset. See the Stimulus/Pulse Phase Diagram for an illustration.

ID and Auto ID check box

Synapse automatically generates a store name. TDT recommends using Auto ID to ensure no store names are duplicated. To make your own store names, clear the **Auto ID** check box.

SCout-1 and SCout-2

Select the radio button in the desired row to feed the parameter to an output line.

Epoch Event Storage

Common Use Cases

Timestamp and store single or multi-channel data when triggered. Use this gizmo to capture behavioral inputs or stimulus parameters to filter and align neurophysiological data.

Data Stored	
Epoc	Event value (optional) and timestamp

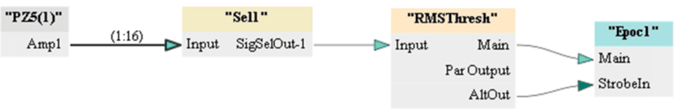
Gizmo Help Slides

Quick Help Slide 1

Epoch Event Storage — Timestamp and store single or multi-channel data when triggered

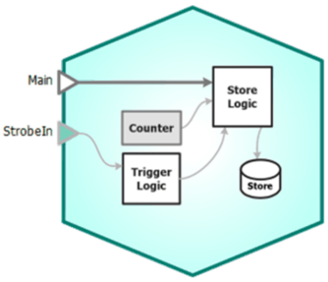
- Capture behavioral inputs and TTLs
- Capture stimulus parameters
- Capture data values on triggered events
- Use timestamps in post-processing to align neurophysiological data with events

Route raw data signals from your neural amplifier to a Unary Processor to threshold an RMS signal. Trigger an Epoch Store of the value if the threshold is crossed.



Main: Single or multi-channel input to store

StrobeIn: Logical input to trigger data saving



None

Note* Digital I/O to RZ or RX processors can be captured, without attaching an Epoc Store, by configuring the Epoc store in the Digital I/O tab of the HAL.

Quick Help Slide 2

Epoch Event Storage

Gizmo Configuration, Data Stored

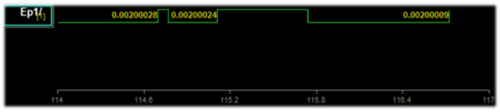
Setup epoch store to work on a timer, with a strobe input, or on a value change.

Internal Timer — Trigger the Epoch Store periodically according to a timer

Strobe Input — Trigger the Epoch Store using an external logic input

Value Change — Trigger the Epoch Store whenever a value on the Main input changes

Ep1/ — Timestamped onset and/or offset event {Epoch}



Reference

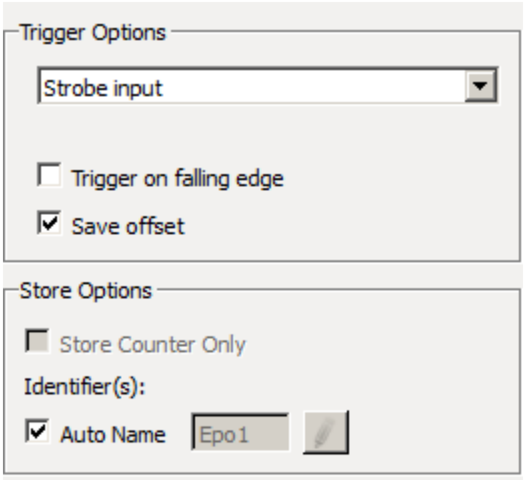
The Epoch Event Storage gizmo stores timestamps and values when triggered. Supports single channel or multiple channel input.

The Runtime Interface

Runtime Plot

An epoch plot is added to the runtime window for visualization that shows the timestamps and values of the stored events.

Epoch Event Configuration Options



Storage Options

Trigger Options

By default, the epoc storage trigger is a **Strobe Input** and requires a logic signal input.

To save a value at a regular interval, change the Trigger Option to **Internal Timer**.

The **Value Change** option stores a timestamp and value whenever the input value changes. If the input signal is multi-channel, then all channels will be stored whenever any channel values changes.

To invert the gizmo input trigger, select **Trigger on falling edge**.

To save timestamps of the onset and offset of the trigger, select **Save offset**.

Check **Store Counter Only** to ignore the Main input and store an incrementing counter value when the gizmo is triggered. This option is only available if the Main input is single channel.

If the Main input contains more than one channel, the additional channels are stored on the same timestamp and given unique identifiers in the data tank.

When using Auto Name, a "/" is appended to the name to indicate when the full epoc is stored (and is not appended when only saving the onset).

Fiber Photometry for RZ10x Processor

Common Use Cases

Real-time control and acquisition of demodulated locked-in amplification signal from any combination of up to 3 light drivers and 2 photosensors on a Lux bank. Can also monitor light power from PM1 from any Lux bank. This is the primary gizmo used in fiber photometry setups using RZ10x processors. Record up to 6 demodulated signals with raw photosensor output and dF/F calculations, too.

Data Stored	
Stream	Raw sensor and demodulated response signals
Stream (optional)	Broadband raw signals
Scaler (optional)	Driver parameters
Outputs	
Calculated signals (optional)	Up to 4 single-channel floats
Timer signal (optional)	Logic signal when LEDs are enabled

Gizmo Help Slides

Quick Help Slide 1

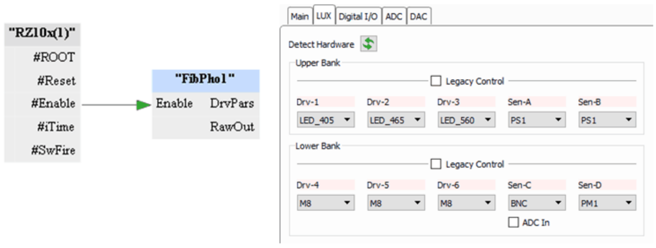
Fiber Photometry for RZ10 — Real-time control and acquisition of demodulated locked-in amplification signal from any combination of up to 3 light drivers and 2 photosensors using the RZ10 integrated LUX components. Includes online light power measurement, stimulus timing controls, and automated fiber bleaching mode.

Multi-fiber fluorescent protein imaging

Drug addiction and behavior

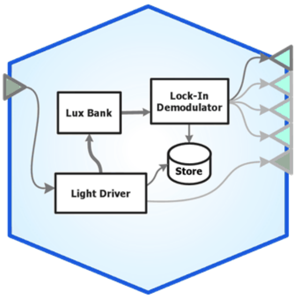
Fear conditioning experiments

RZ10(x) is tied directly to the FibPho gizmo. Detected LUX hardware on the RZ10(x) is controlled within the gizmo



Enable: Master enable for LEDs. Typically tied to #Enable

Lux Bank: Detected hardware (RZ10 → LUX → Detect Hardware) is automatically assigned to Light Driver and Sensor tabs in the gizmo



**-Res:* Not shown. Demodulated signals are saved by default but can be added as outputs for further processing
StrobeOut: Not shown. Logic timing signal when the Timing Control feature is enabled

Quick Help Slide 2

Fiber Photometry for RZ10

User Interface, API, Data Stored

Dynamic control of demodulated signal filtering (**Sensors**), LED driver settings (**Drv-{N} [xxx]**). Toggle Power Meter and Fiber Bleaching modes during Preview Mode (**Display Control**)

*DriveFreq** — Control the frequency of each driver waveform {Read/Write}

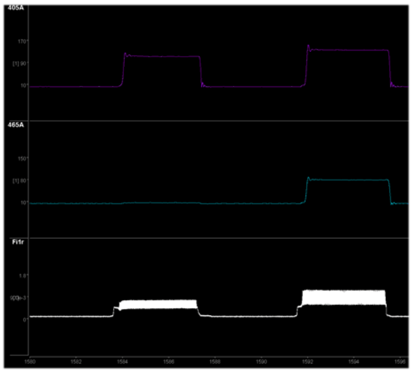
*DriveLevel** — Control the level of each driver waveform {Read/Write}

*DriveOn** — Turn each driver waveform on and off {Read/Write}

Lowpass — Control the lowpass filter setting for demodulated signals {Read/Write}

Driver x Sensor — Demodulated data stream calculated via locked-in amplification. Each signal will contain information about the fluorescent protein response to one wavelength of light {Stream}

Fi1r — Contains ADC information about raw photosensor data input {Stream}



Quick Help Slide 3

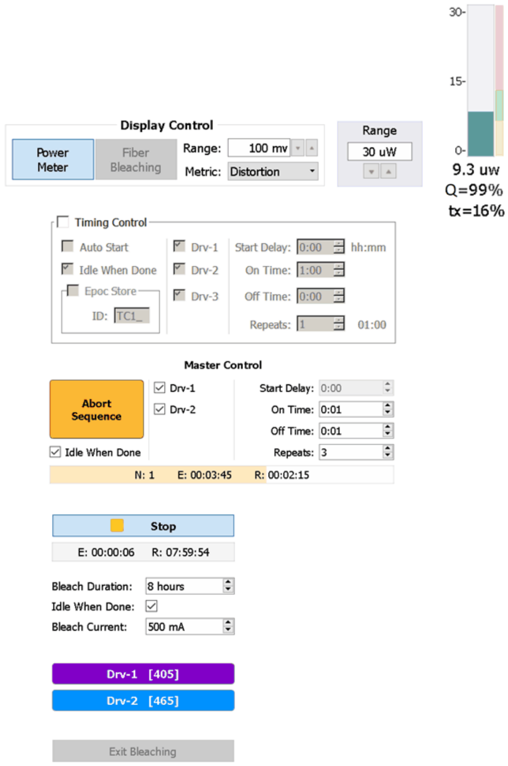
Fiber Photometry for RZ10

Special Runtime Features

Power Meter – Display measured power and transmission percentage of LED to subject fiber from connected PM1. Only available in Preview Mode

Timing Control – Cycle the LEDs On/Off for set durations and repeats during Run-Time. Enable the Timing Control during Design-Time, then activate and adjust the sequence as needed during Run-Time in Master Control

Fiber Bleaching – Set a constant current/ light output on connected LEDs for a set duration to bleach patch cables. The system will go to Idle mode when the timer if finished. Only available in Preview Mode



Reference

The Fiber Photometry gizmo includes designtime and runtime control for the RZ10x LuxIO processor built-in LED light drivers. It uses lock-in amplification to measure the real-time power of the resulting fluorescence response at the LED driving frequencies.

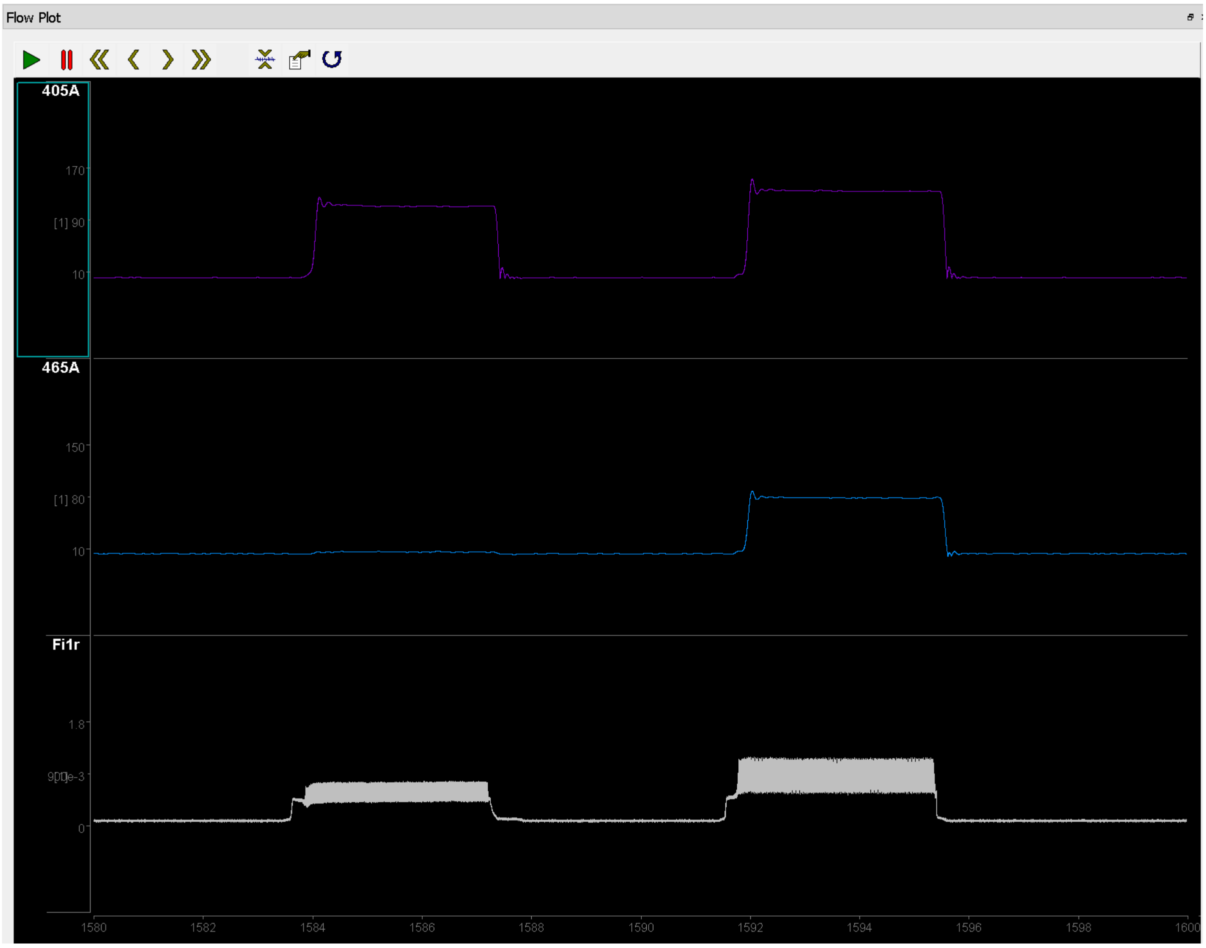
It provides an estimate of the LED power output and can also measure power at the subject end. It has a built-in timer for timed stimulus control and automatic photo-bleaching capabilities.

Each Fiber Photometry gizmo targets a single LUX bank of components on the RZ10x, controlling up to three LED drivers and reading two photosensors. PM1 power meters are handled uniquely; they can be accessed from any LUX bank.

The Runtime Interface

Runtime Plot

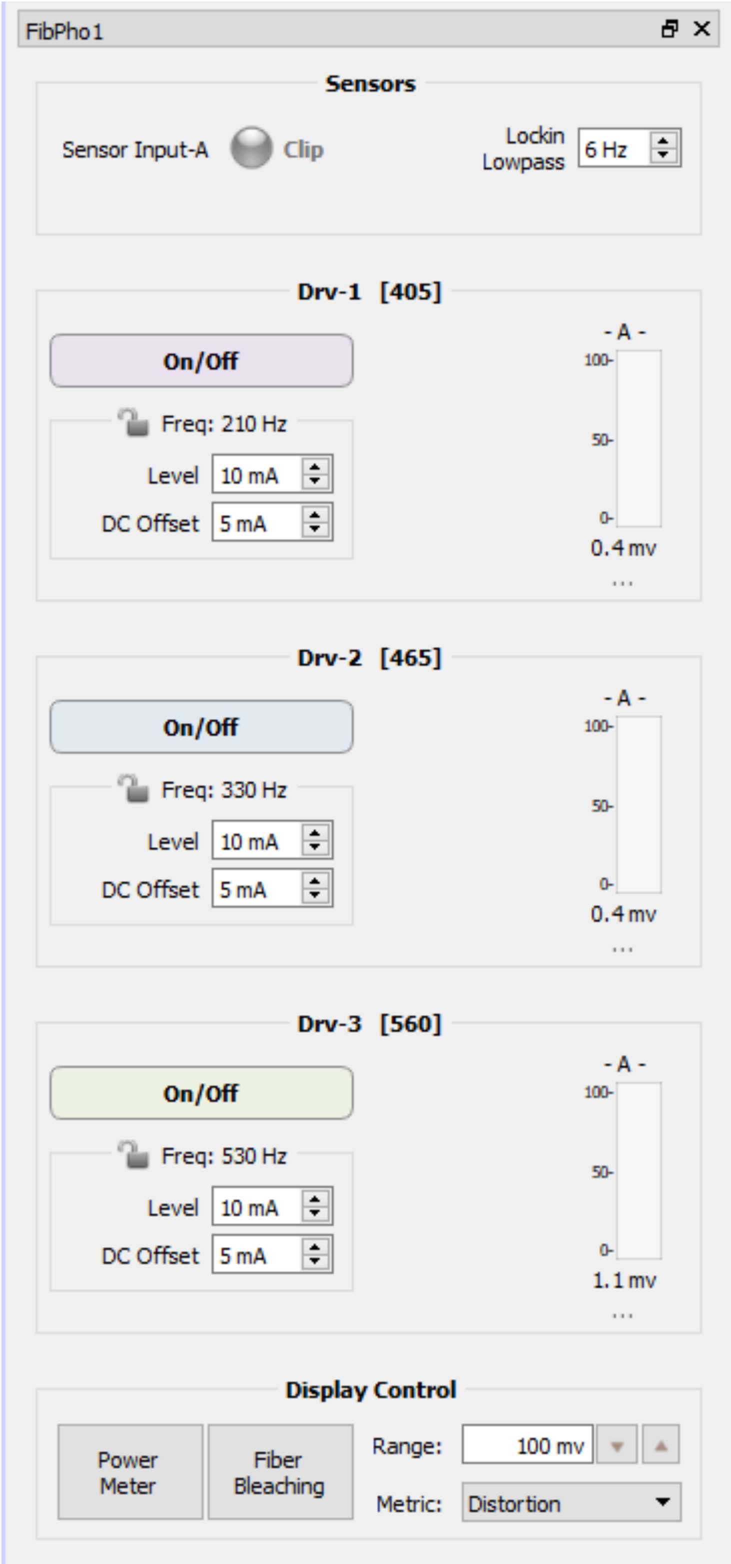
A plot is added to the runtime window for visualization.



Flow Plot Showing Demodulated Responses

The subplots shown in a runtime plot represent data storage you chose in the designtime options. In the example above, the streamed data shows the resulting power output (405A, 465A) at the frequencies of the 405 and 465 driver signals on the sensor A input. The Fiber Photometry gizmo also stores and displays broadband raw input signals and driver parameters, depending on selections made at designtime. Fi1r is the raw sensor input in this example.

Runtime Controls

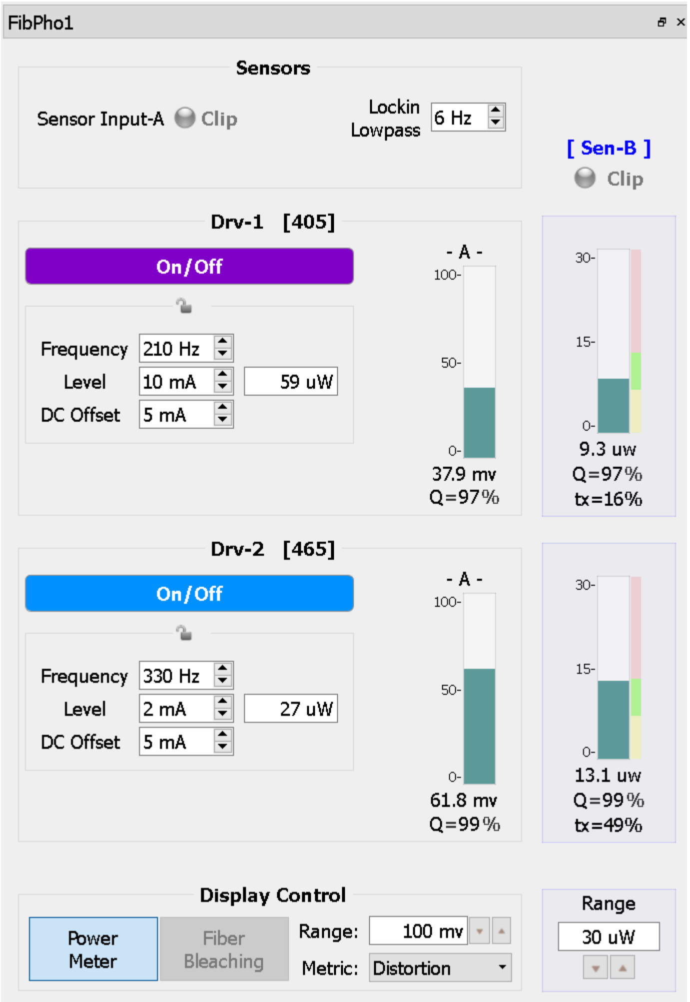


Runtime Interface

Note that the Power Meter and Fiber Bleaching modes are only available in Preview mode. Power Meter mode is for system verification, and Fiber Bleaching is a special feature that doesn't involve data collection, so these modes can't be accidentally enabled during an experiment (Record mode).

Sensor(s) Options	Description
Clipping Indicator(s)	An indicator for each sensor flashes when the user-defined clipping threshold is passed. The clipping indicator will also light up if the input voltage is below 10 uV to indicate a bad connection. If a PM1 is detected, the clipping indicator for the PM1 will only be active when 'Power Meter' in the Display Control settings is enabled.
Lockin Lowpass	Control the lowpass filter that determines the bandwidth around each demodulation frequency used for the power calculation
Drv-{n} Options	
On/Off	A button enables the light driver and indicator when the light is on. If the driver has the name of the LED wavelength then this button color will match the LED wavelength.
Parameters	Knobs and value entry boxes allow runtime control of light driver Frequency, Level, and DC Offset parameter values. The 'Lock Freqs at Runtime' setting disables the Frequency control, and the 'Auto-Calc Offsets' setting disables the DC Offset control. If Launch Power Est is set in the Light Driver(s) Tab during designtime, then the estimated light power output will be displayed next to the Level.
Results	A result for each sensor is dynamically displayed as a single value in millivolts
Display Control	
Range	Set the range of the response bar indicator
Metric	Show an optional metric beneath the response bar. Distortion measures the amount of signal distortion in the LED output signal relative to a pure sine wave at the set frequency. Distortion greatly impacts the demodulation measurement because it affects the frequency characteristics of the driving signal. While you want to keep the driving current low, you also want to make sure the distortion is low. This measure is shown as a Quality score on the runtime display and should ideally be >95%. During system setup use the Level and DC Offset settings to adjust this. S/N gives a measure of the signal relative to noise measured at harmonics of the lockin lowpass frequency.

Power Meter

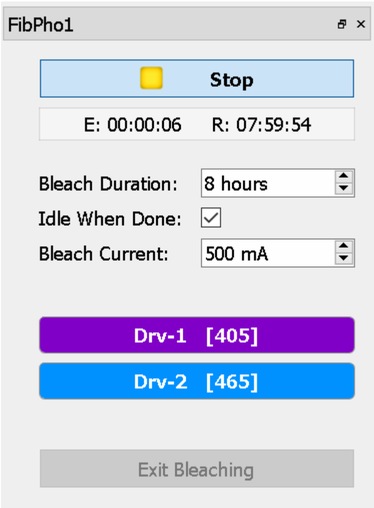


Runtime Interface for Power Meter Mode

Note that the Power Meter mode is only available when a LUX PM1 is detected in the RZ10x. This mode is for system verification and is only available in Preview mode, so it can't be accidentally enabled during an experiment (Record mode).

Sensor Options	Description
Clipping Indicator	The clipping indicator for the PM1 is displayed above the column of power indicators.
Drv-{n} Options	
Indicators	The measured power level at the driver frequency is shown next to each driver. The DC contribution of the individual driver is included in the power measurement for that driver. The colored bars to the right of the indicator show the designtime Target Range. The signal quality relative to a pure sine wave is displayed under the measured power. If Launch Power Est is set in the Light Driver(s) Tab during designtime, then the total signal transmission through the system (tx) is also shown.
Display Control	
Range	Sets the maximum range of the power meter indicators

Fiber Bleaching



Runtime Interface for Fiber Bleaching Mode

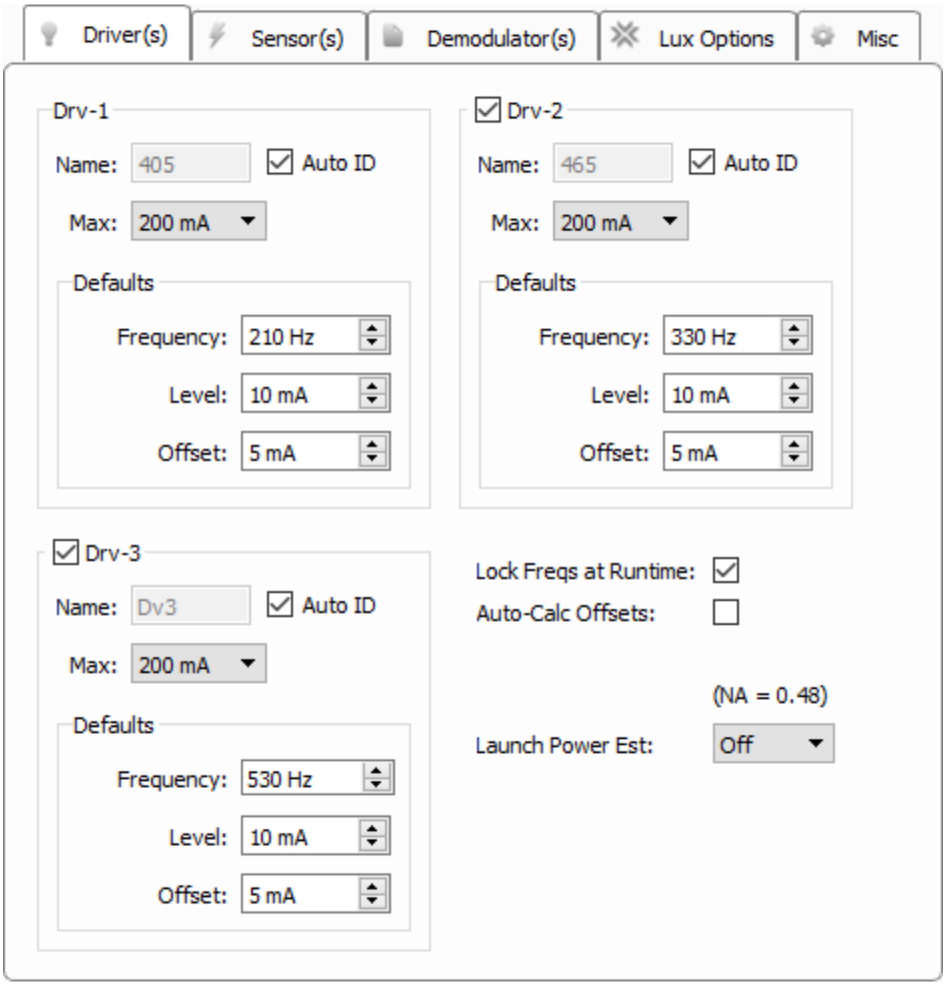
The Fiber Bleaching mode is a special feature for driving high light power through the cables for an extended period of time to reduce the cable's autofluorescence and increase signal to noise before running an experiment. This mode doesn't involve data collection so it is only available in Preview mode, so these modes can't be accidentally enabled during an experiment (Record mode).

Check with your fiber optic cable manufacturer for recommendations on fiber bleaching current, duration, and how often this should be done.

Setting	Description
Start/Stop	Use the Start/Stop button to begin or end the bleaching timer. The elapsed (E) and remaining (R) time are shown in the progress bar.
Idle When Done	Check this box to automatically switch the system to Idle mode when the bleaching timer is done
Bleach Current	Set the desired bleach current. Note that the current set here in Fiber Bleaching mode temporarily overrides the maximum current specified in the Light Driver(s) Tab during designtime.
Drv-{n}	Select which light drivers to enable during bleaching

Fiber Photometry Configuration Options

Driver(s) Tab



Driver(s) Tab

Drv-{n} Options	Description
Name	Three characters to identify the driver. If using integrated LUX LEDs, this defaults to the detected LED wavelength. If the name is the wavelength, the runtime button to enable the driver and runtime demodulation plots will be colored to match this wavelength. The name will be the first three characters in any demodulated store based on this driver signal.
Max	Set the current output range as low as possible to match your desired driver signal. For the 1000 mA, 500 mA, and 200 mA settings, the actual driver hardware precision is adjusted to maximize dynamic range. The 50 mA setting uses the same driver precision as the 200 mA setting but allows you to adjust the Level/Offset in 0.1 mA increments at runtime, instead of the default 1 mA increment. This is useful if you are using a larger core diameter fiber on the LED outputs and need finer tuning to achieve lower signal output. In general, 200 mA is the recommended setting.
Defaults Options	
Frequency	Modulation frequency of the light source sine wave.
Level	Peak-to-peak light source sine wave amplitude.
Offset	DC shift added to the output sine wave. The offset is adjusted to reach the linear range of the physical light driver output and minimize signal distortion. More on that later.

Lock Freqs at Runtime

Uncheck this for debugging or system setup, otherwise leave it unchecked to prevent accidental frequency changes during recording.

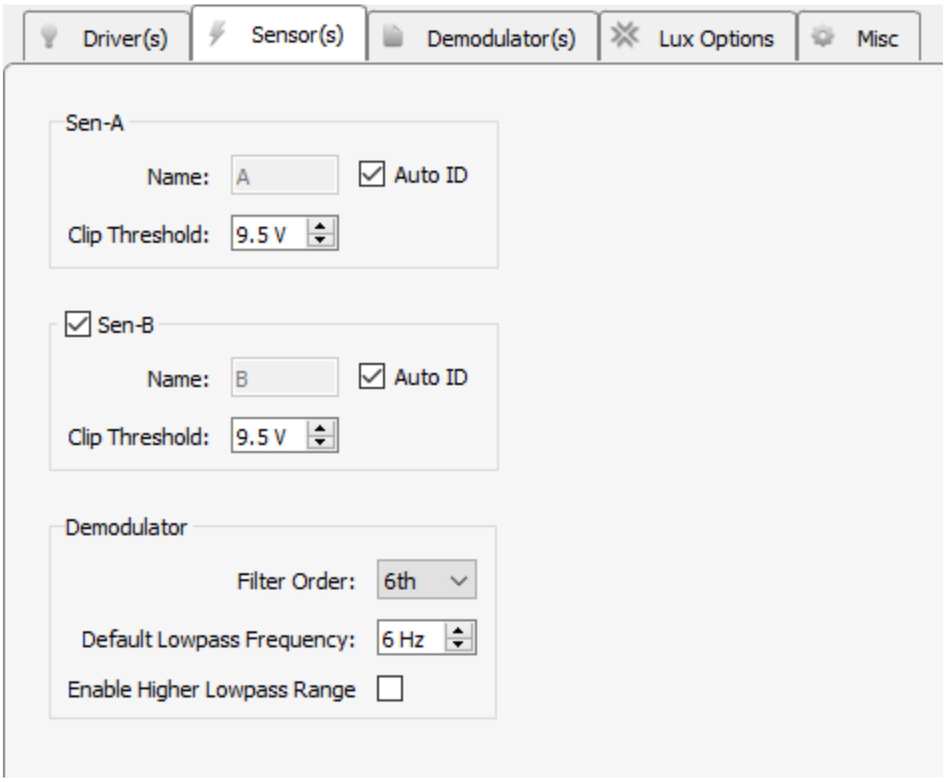
Auto-Calc Offsets

Automatically sets the Offset to 10% of the Level (rounded up) at runtime, with a minimum of 5 mA and maximum of 20 mA. This can reduce signal distortion at higher Level settings.

Launch Power Est

This option will display an estimate of the LED power output (in uW) at runtime through the selected fiber core diameter connected to the LED, assuming the cable Numerical Aperture is 0.48. Use this setting with a PM1 power meter to get an estimation of overall light transmission through the entire optical chain.

Sensor(s) Tab

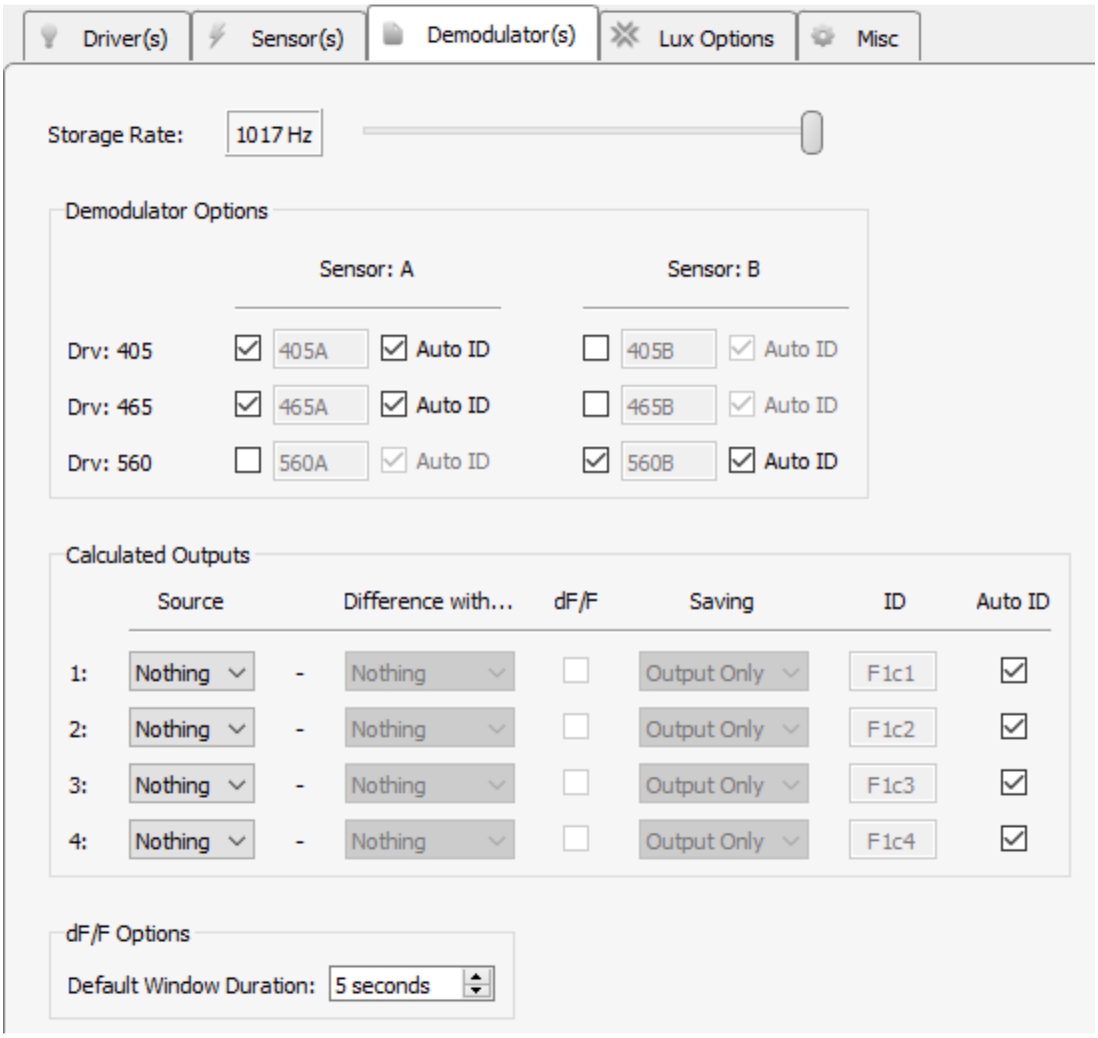


Sensor(s) Tab

Note: PM1 power meters do not appear in this tab but are available in the runtime interface in Preview mode.

Sen-{x} Option	Description
Name	Sensor name as it will appear in the runtime interface. This defaults to A/B for the upper LUX bank, and C/D for the lower LUX bank. The first letter of the sensor name is used as the last letter of the demodulated store name.
Clip Threshold	The runtime interface alerts the user if the raw A/D sensor input voltage value goes beyond this value. For the PS1 LUX sensor, this is set to 9.5 V. Other external photosensors may have a different clipping threshold.
Demodulator Option	
Filter Order	Higher order filters tighten the band around the response frequency
Default Lowpass Frequency	Determines the band around the frequency of interest to do the demodulation calculation. This can be modified at runtime.
Enabled Higher Lowpass Range	Check this box to raise the lowpass frequency limit to 100 Hz for TEMPO (voltage sensor photometry) where sensors have a wider bandwidth. See the Fiber Photometry User Guide for more information.

Demodulator(s) Tab



Demodulator(s) Tab

Storage Rate

Set the storage sampling rate for the demodulated and calculated signals configured on this tab.

Demodulator Options

Use the matrix of check boxes to select the combinations of sensors and drivers that will be used for demodulation. All available sensor signals can be demodulated against all light driver signals if desired.

Calculated Outputs

Perform up to four real-time calculations on the demodulated data streams. Choose a **Source** demodulated signal, optionally subtract another demodulated signal in the **Difference with...** column, and optionally perform a delta F over F calculation on it (dF/F checkbox) to compare the strength of the signal relative to the baseline.

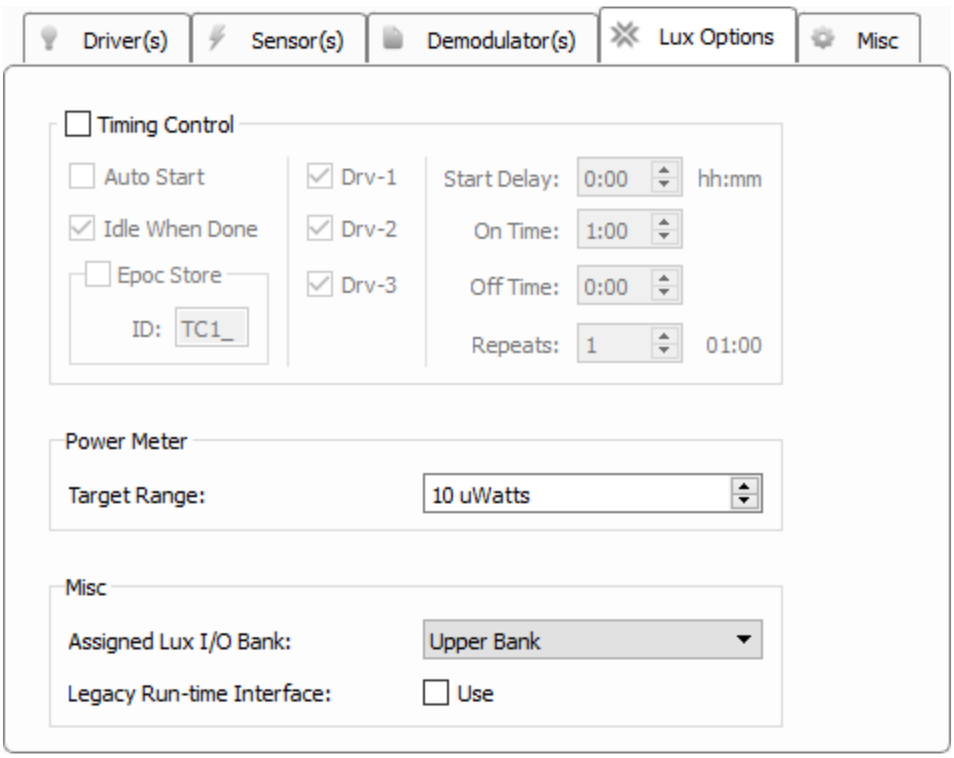
Saving

The calculated signal is available as a gizmo output so other gizmos can attach to it and do further processing in real-time. The calculated signal can also be optionally shown on the Flow Plot at runtime, and optionally be saved to disk as well.

dF/F Options - Default Window Duration

Sliding average window size used as the baseline for the delta F over F computation.

Lux Options Tab



Lux Options Tab

Timing Control

Timing control options are used to cycle the LEDs On and Off for set durations and repeats during runtime. This feature is for running long (greater than 1 hour) experiments where photobleaching becomes a concern.

Timer Option	Description
Auto Start	Automatically start the timer when entering runtime mode. Note that there is ~1 second delay after the start of recording before the timer begins.
Idle When Done	Automatically return to Idle mode when the timer has finished
Epoc Store	Store the onset/offset timestamps of when the driver signals were active
Drv-{n}	Decide which drivers the timing control can control

Power Meter

This option group is only available if a PM1 is detected in the RZ10x.

Power Meter Option	Description
Target Range	Determines where to place the visual indicator next to the power meter at runtime for testing overall system power. 75% to 133% of this target value is color green. Anything below that range is colored yellow, and anything above is colored red.

Misc

Misc Option	Description
Assigned Lux I/O Bank	Only one gizmo can target each horizontal Lux I/O bank. Note: PM1 power meters are unique in that they can be accessed from any Fiber Photometry gizmo regardless of their physical bank location. So if you have an RZ10x you don't need to physically move the PM1 to measure power from both banks.
Legacy Run-time Interface	Use the runtime interface from the Fiber Photometry gizmo for non-RZ10 devices

Misc Tab

Misc Tab

Required Sample Rate

Tells the RZ what minimum sample rate this gizmo requires. Typically 6K is enough. Only increase this if the driver frequency needs to go beyond 1-2 kHz for your experiment, which is rarely done.

Drivers On at Runtime

Check this option if you want to automatically turn on the LED drivers when switching to Preview or Record mode.

Store Driver Signals

Stores the signals used to drive the LEDs at the RZ10x system rate. The store name is the first two letters of the gizmo name, followed by the last letter of gizmo name, followed by 'd' (default 'Fi1d'). One channel per light driver.

Store Driver Parameters

All light driver parameters are timestamped and stored to disk two seconds after a change has been made to any of the driver parameters in the runtime user interface.

Store Sensor Signals

Stores the raw sensor signals at the RZ10x system rate. The store name is the first two letters of the gizmo name, followed by the last letter of gizmo name, followed by 'r' (default 'Fi1r'). One channel per sensor.

Fiber Photometry for non-RZ10x Processors

Common Use Cases

Real-time control and acquisition of demodulated locked-in amplification signal from any combination of up to 4 light drivers and 2 photosensors. This is the primary gizmo used in fiber photometry setups using processors other than RZ10x. Record up to 8 demodulated signals with raw photosensor output too.

Data Stored	
Stream	Demodulated response signals
Stream (optional)	Broadband raw signals
Scaler (optional)	Driver parameters
Outputs	
Driver output voltage (optional)	Up to 4 single-channel floats
Calculated signals (optional)	Up to 4 single-channel floats

Gizmo Help Slides

Quick Help Slide 1

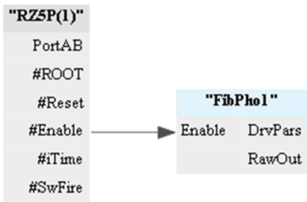
Fiber Photometry — Real-time control and acquisition of demodulated locked-in amplification signal from any combination of up to 4 light drivers and 2 photosensors

Multi-fiber fluorescent protein imaging

Drug addiction and behavior

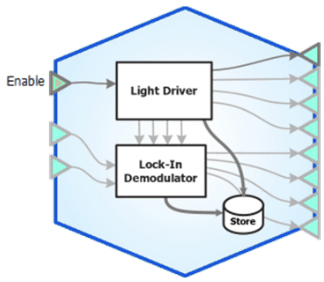
Fear conditioning experiments

RZ5P is tied directly to the FibPho gizmo. DAC inputs and ADC outputs are controlled within the gizmo



Enable: Master enable for LEDs. Typically tied to #Enable

Sensor In-:* Not shown. Typically ADC inputs are used, and are set within the gizmo



**-Drv:* Not shown. Typically driver signals go right to the DAC outputs, and are set within the gizmo

**-Res:* Not shown. Demodulated signals are saved by default but can be added as outputs for further processing

Quick Help Slide 2

Fiber Photometry

User Interface, API, Data Stored

Dynamic control of LED driver settings and demodulated signal filtering

*DriveFreq** — Control the frequency of each driver waveform {Read/Write}

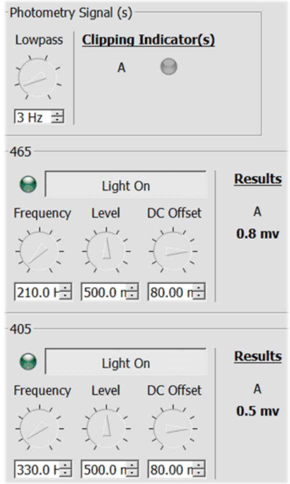
*DriveLevel** — Control the level of each driver waveform {Read/Write}

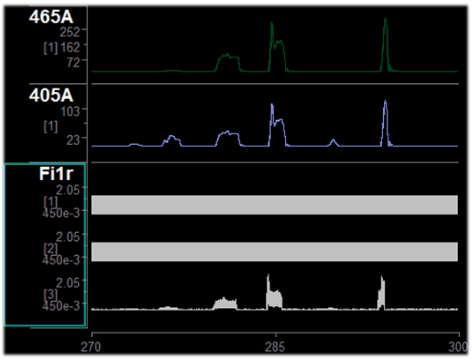
*DriveOn** — Turn each driver waveform on and off {Read/Write}

Lowpass — Control the lowpass filter setting for demodulated signals {Read/Write}

Driver x Sensor — Demodulated data stream calculated via locked-in amplification. Each signal will contain information about the fluorescent protein response to one wavelength of light {Stream}

Fi1r — Contains DAC/ADC information about LED driver signal outputs and raw photoreceiver data input {Stream}





Reference

The Fiber Photometry gizmo, when used with processors that are not the RZ10x, includes designtime and runtime control of up to four voltage output signals to drive external light drivers. It uses lock-in amplification to measure the real- time power of the resulting fluorescence response at the light driving frequencies. It controls up to four light driver signals and reads two photosensors via the RZ analog inputs.

The Runtime Interface

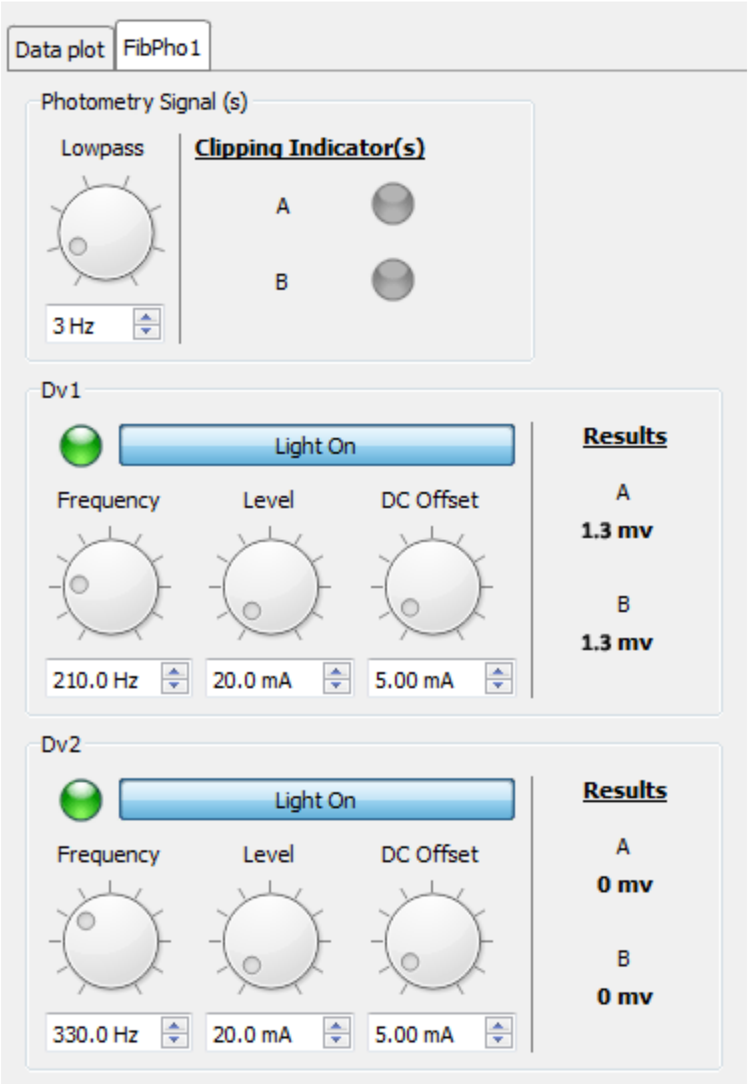
Runtime Plot

A plot is added to the runtime window for visualization.

Flow Plot Showing Demodulated Responses

The subplots shown in a runtime plot represent data storage you chose in the designtime options. In the example above, the streamed data shows the resulting power output (such as Dv1A) at the frequency of interest when comparing the selected driver (such as Drv1) to the selected sensor input (such as sensor A). Simultaneous neural recordings from a different gizmo are integrated in the plot for a quick visual comparison. The Fiber Photometry gizmo also stores and displays broadband raw input signals and driver parameters, depending on selections made at designtime.

Runtime Controls



Runtime Interface

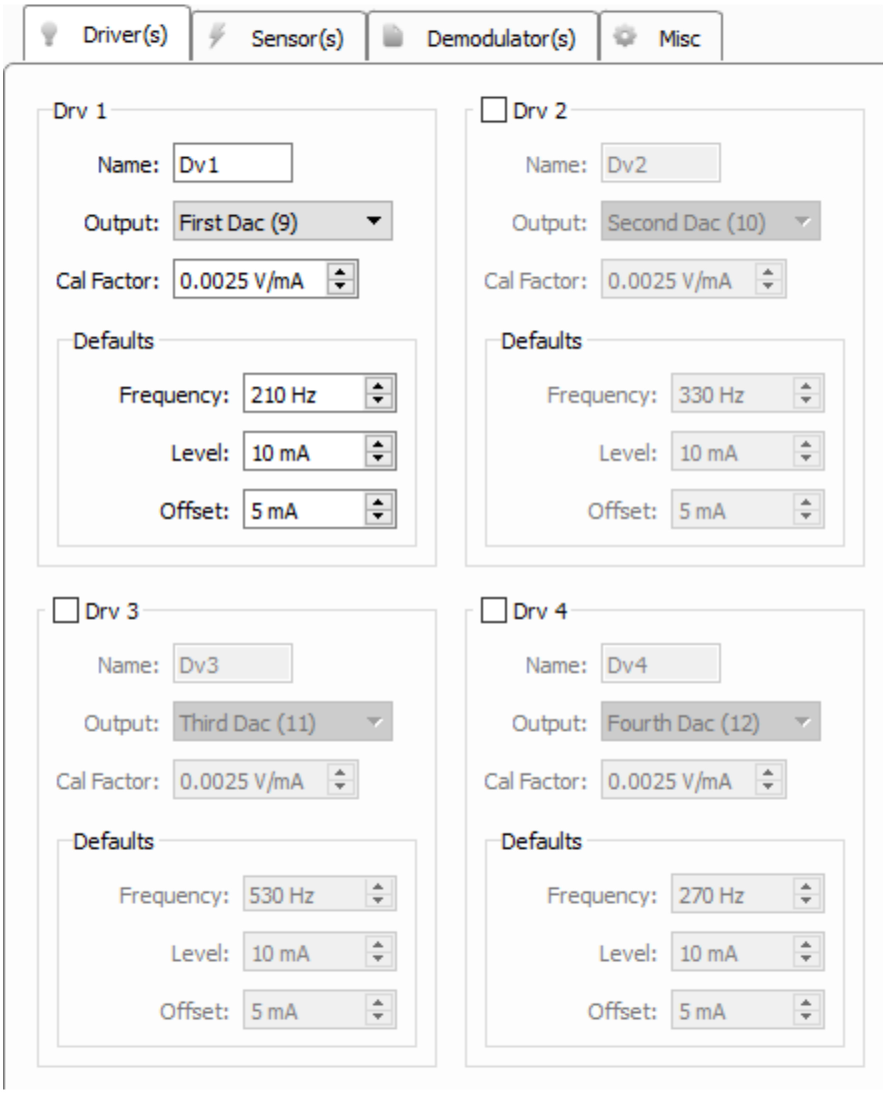
The runtime window includes:

Options	Description
Clipping Indicator(s)	Two indicators, one for each sensor, flash when the user-defined clipping threshold is crossed. The clipping indicator will also light up if the input voltage is below 10 uV to indicate a bad connection.
Lowpass Filter	A knob and value entry box that controls the lowpass filter that determines the bandwidth around each demodulation frequency used for the power calculation
Dv{x} Options	
Light On	A button enables the light driver and an indicator is lit green when the light is on
Parameters	Knobs and value entry boxes allow runtime control of light driver Frequency, Level, and DC Offset parameter values
Results	A result for each sensor is dynamically displayed as a single value in millivolts

Fiber Photometry Configuration Options

See The Options Area and Templates for more information on the gizmo name, source, global options, and displaying the block diagram.

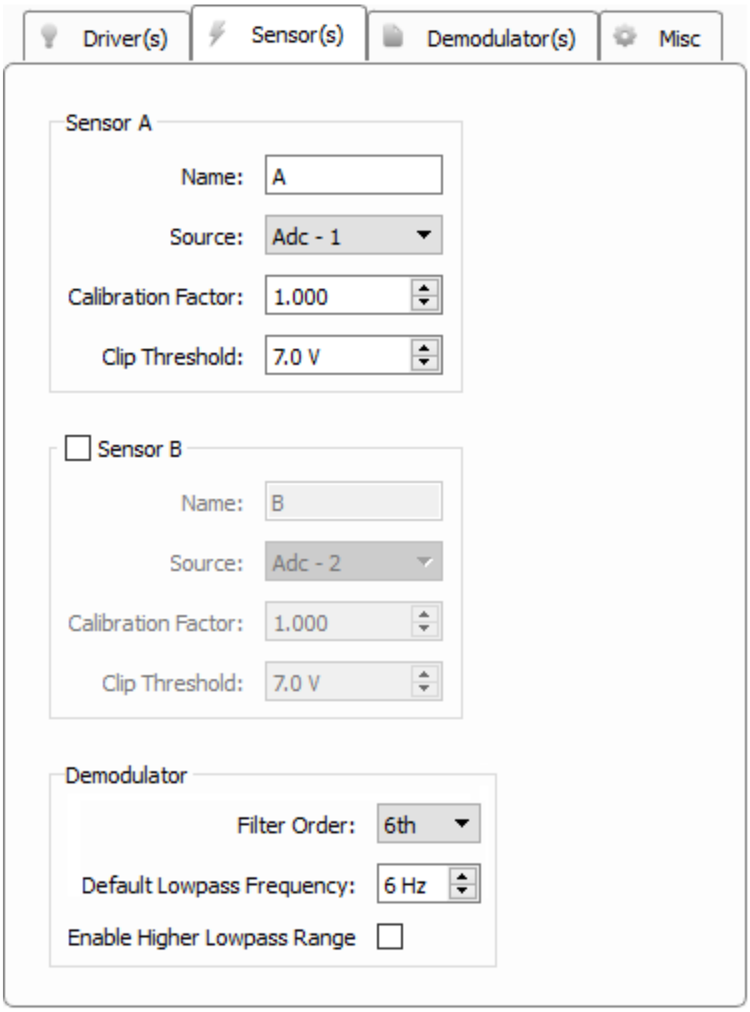
Driver(s) Tab



Driver(s) Tab

Drv-{n} Options	Description
Name	Three characters that serve as the first three characters in the store name.
Output	Specify the DAC number (channel number in parenthesis), or send this as a gizmo output.
Cal Factor	Scale factor used to convert desired milliamps into voltage.
Defaults Options	
Frequency	Modulation frequency of the light source sine wave
Level	Peak-to-peak light source sine wave amplitude
Offset	DC shift added to the output sine wave. The offset is adjusted to reach the linear range of the physical light driver output and minimize signal distortion. More on that later.

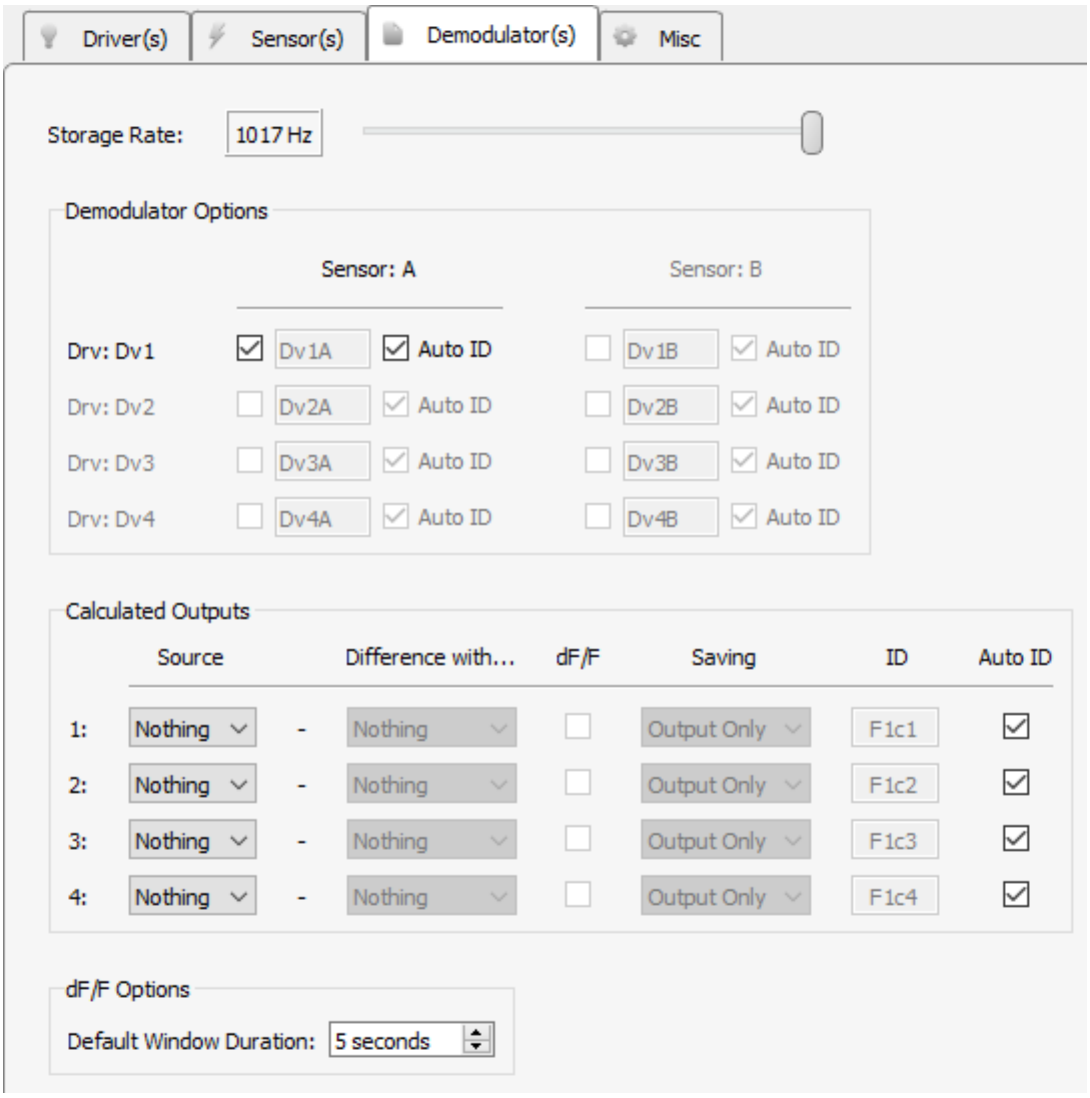
Sensor(s) Tab



Sensor(s) Tab

Sensor Option	Description
Name	Give the sensor a name as it will appear in the runtime interface. The first letter of the sensor name is used as the last letter of the streaming data store name.
Source	The sensor input can be an analog input from the front panel of the hardware or a floating point signal output of another hardware source or gizmo.
Calibration Factor	Scales the sensor data
Clip Threshold	Raw A/D sensor input voltage value to light runtime indicator (no calibration factor applied)
Sensor B check box	Enables a second sensor input
Demodulator Option	
Filter Order	Higher order filters tighten the band around the response frequency
Default Lowpass Frequency	Determines the band around the frequency of interest to do the demodulation calculation. This can be modified at runtime.
Enabled Higher Lowpass Range	Check this box to raise the lowpass frequency limit to 100 Hz for TEMPO (voltage sensor photometry) where sensors have a wider bandwidth.

Demodulator(s) Tab



Demodulator(s) Tab

Storage Rate

Set the storage sampling rate for the demodulated and calculated signals configured on this tab.

Demodulator Options

Use the matrix of check boxes to select the combinations of sensors and drivers that will be used for demodulation. All available sensor signals can be demodulated against all light driver signals if desired.

Calculated Outputs

Perform up to four real-time calculations on the demodulated data streams. Choose a **Source** demodulated signal, optionally subtract another demodulated signal in the **Difference with...**

column, and optionally perform a delta F over F calculation on it (dF/F checkbox) to compare the strength of the signal relative to the baseline.

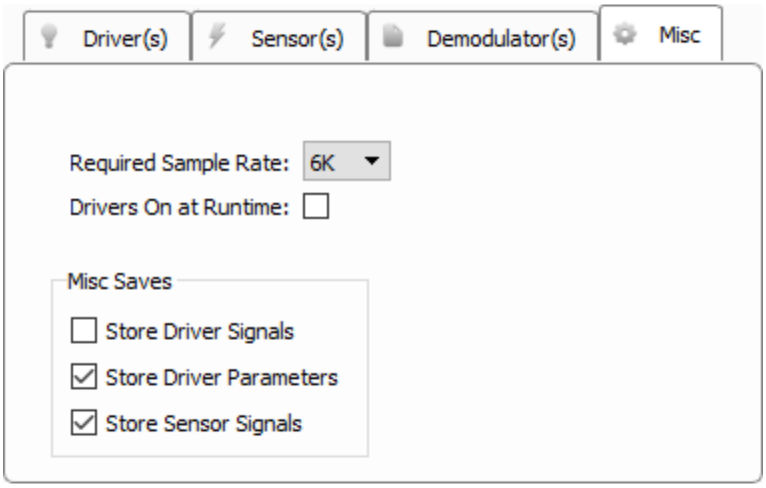
Saving

The calculated signal is available as a gizmo output so other gizmos can attach to it and do further processing in real-time. The calculated signal can also be optionally shown on the Flow Plot at runtime, and optionally be saved to disk as well.

dF/F Options - Default Window Duration

Sliding average window size used as the baseline for the delta F over F computation.

Misc Tab



Misc Tab

Required Sample Rate

Tells the RZ what minimum sample rate this gizmo requires. Typically 6K is enough. Only increase this if the driver frequency needs to go beyond 1-2 kHz for your experiment, which is rarely done.

Drivers On at Runtime

When selected, output driving signals are "on" when recording begins. Otherwise, light drivers must be turned on manually in the runtime interface.

Store Driver Signals

Stores the signals used to drive the LEDs at the RZ system rate. The store name is the first two letters of the gizmo name, followed by the last letter of gizmo name, followed by 'd' (default 'Fi1d'). One channel per light driver.

Store Driver Parameters

All light driver parameters are timestamped and stored to disk two seconds after a change has been made to any of the driver parameters in the runtime user interface.

Store Sensor Signals

Stores the raw sensor signals at the RZ system rate. The store name is the first two letters of the gizmo name, followed by the last letter of gizmo name, followed by 'r' (default 'Fi1r'). One channel per sensor.

File Stimulation

Common Use Cases

Play custom waveforms from a list of files on disk, which includes WAV files and MAT files. Use this for speech studies, psychoacoustics, or for custom audio or electrical stimulus presentations.

Data Stored	
Epoc (optional)	Parameter values when triggered
Stream (optional)	Raw stimulus waveform
Outputs	
Stim	Stimulus waveform, single channel floating point
StimSync	Logic signal when file stimulation is active
Par Output	Full parameter stream

Gizmo Help Slides

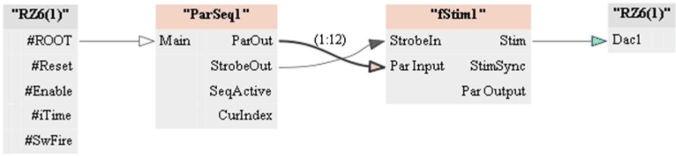
Quick Help Slide 1

File Stimulation — Play custom waveforms from a list of files on disk, which includes WAV, F32, and MAT files

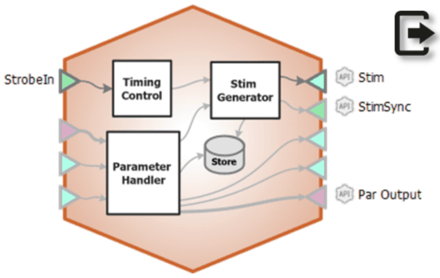
Present custom electrical stimulation patterns

Social experiments with mouse and rat vocalizations

RZ6 processor uses Parameter Sequencer to control uStim1 gizmo for stimulus presentation. Signal sent to DAC1 on RZ6 for play out of a speaker.



StrobeIn: Optional input to control presentation timing



Stim: Single-channel stimulation signal

StimSync: Logic signal that is high during a stimulus presentation

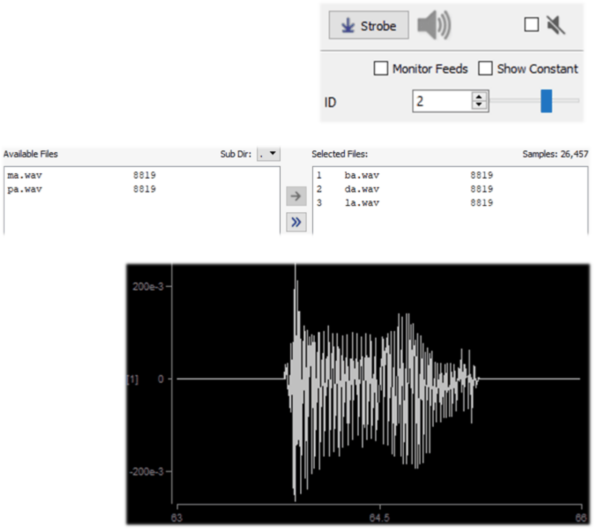
ParOut: Parameter values for the given presentation

Quick Help Slide 2

File Stimulation

User Interface, API, Data Stored

Manually strobe and mute stimuli presentations. Change file ID during runtime



Mute — Turn the stimulus on & off during a presentation {Read/Write}

Strobe — In Manual mode, strobe the gizmo to give a stimulus presentation {Write}

StimActive — Logic high during a stimulus presentation {Read}

fS1r — Raw output waveform {Stream}

fS1p — All parameters, stored when stimulus is triggered {Strobe}

Quick Help Slide 3

File Stimulation

Parameter Table

This gizmo has a Parameter Table to dynamically control values its parameters from other gizmos and during runtime

Parameter Tables provide the user dynamic control over parameter values. Each one of the Modes will change how the user can control and interact with the parameter.

Param In – Dynamically control parameter values from Parameter Sequencer or Parameter Manifold

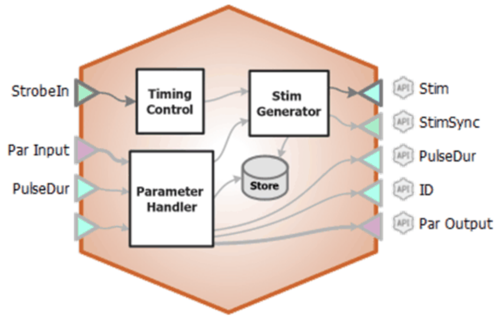
Constant – The value is immutable at runtime

Scalar In – Pass a scalar value from another gizmo for dynamic real-time control

Widget – Creates a widget (slider) on the user interface that is controlled manually by the user or through the API.

Scalar Out (Scout) – Makes this value available as a gizmo output to connect to another gizmo input in real-time

Epoc – Save the parameter values used for each stimulus presentation



Changing the Mode of parameters will change the input and output options on the gizmo.

All parameters can be read through API. Only parameters in Widget mode can be written to.

All parameters are connected to the "Par Output" gizmo output link

File Stim Parameters: Show All

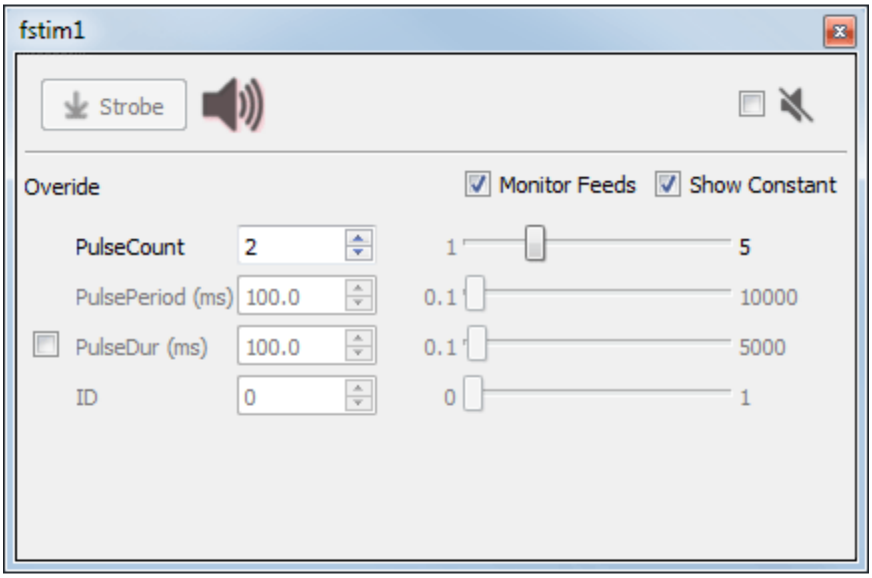
	Name	Mode	Value	Jit(%)	Min	Max	Epoc	ID	Auto ID	SCout-1
2	PulseCount	Constant	1	0.0	1	10000	None	Pcnt	☑	○
3	PulsePeriod (ms)	Widget	100.0	0.0	0.1	10000.0	None	Pper	☑	○
4	PulseDur (ms)	Scalar In-1	100.0	0.0	0.1	100000000	None	Pdur	☑	◉
5	ID	Param In	1	0.0	0	3	None	WVid	☑	○

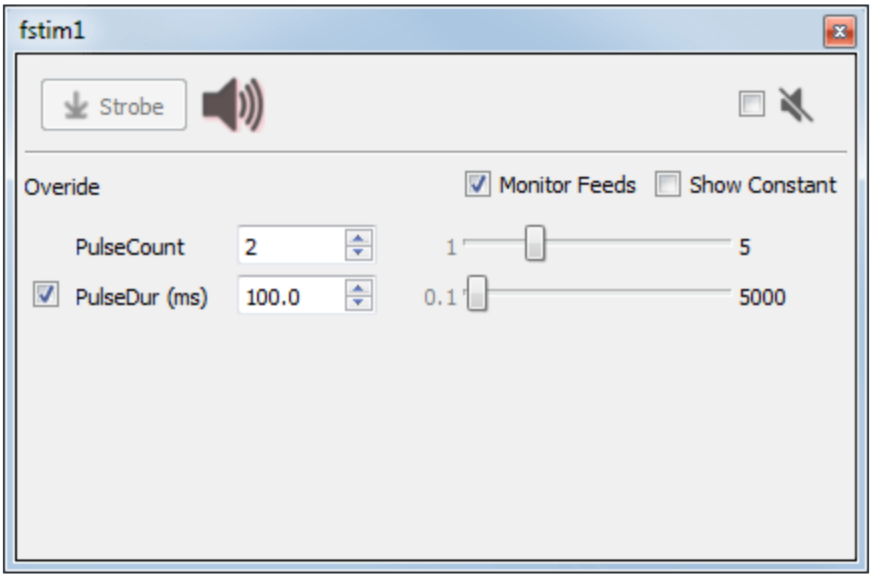
Reference

The File Stimulation gizmo plays stimulus waveforms from a list of files. It supports timing control and dynamic parameters.

File Stimulation Runtime Interface

If enabled in the gizmo configuration, a control tab is added at runtime. Parameters that can be controlled dynamically are shown in black (active). You can enter a value in the field, use up and down arrows, or drag a slider to modify to parameter value. You can show only the elements you need or hide the entire control.





Two Versions of the File Stimulation Runtime Tab

The illustrations above, show two version of the floated tab, one with all the parameters shown and one with only the runtime widget controlled parameter and the **Param In** line (with override selected) shown.

Click and release the **Strobe Button** to trigger a manual strobe pulse.

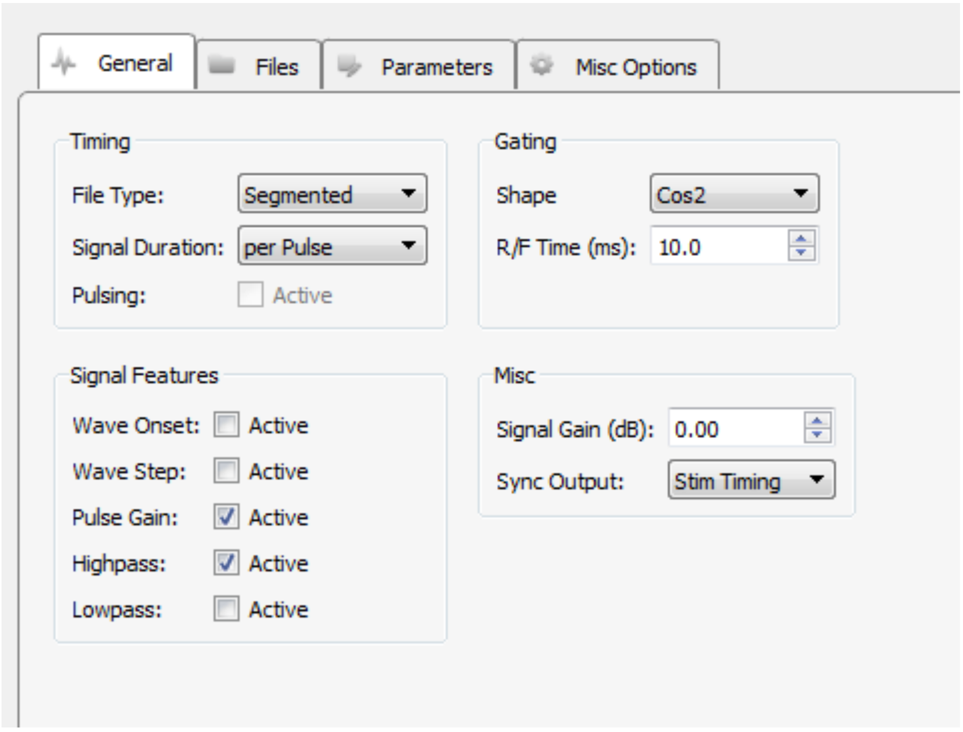
Select the **Mute Button** check box to zero the stimulus signal.

Select the **Monitor Feeds** check box to show stimulus parameters controlled by an input signal. Also adds an Override column and check box to the left. Select the **Override** check box to adjust the parameter value manually instead of using the input signal.

Select the **Show Constant** check box to display values for parameters set to Constant. They will appear gray.

File Stimulation Configuration Options

General Tab



General Tab

Timing

You can choose to use the whole file or segments of the file. When using file segments, you can choose to set the duration of the stimulus waveform per Pulse, per Parameter, or per Strobe. You may also have to define the onset, or starting point of the segment, and step size in the parameter table. The step size allows you to use every nth sample in the signal.

When the **Pulsing Active** check box is selected, pulse duration and pulse count parameters are enabled in the parameter table and the stimulus is triggered when the strobe goes high, the pulse parameters are then followed and the stimulus ends when the pulse count is met or the strobe goes low. The next stimulus is triggered by the next strobe input.

Gating

Gates serve to attenuate the signal during the onset and offset of the signal, gaining or decreasing in intensity, for the purpose of removing onset/offset related artifacts from this signal. You can choose one of several common gate shapes to apply to the signal. The **R/F Time** defines the length of time over which the gate is applied, therefore, the length of time in which the signal goes from 0 to full signal strength or visa-versa.

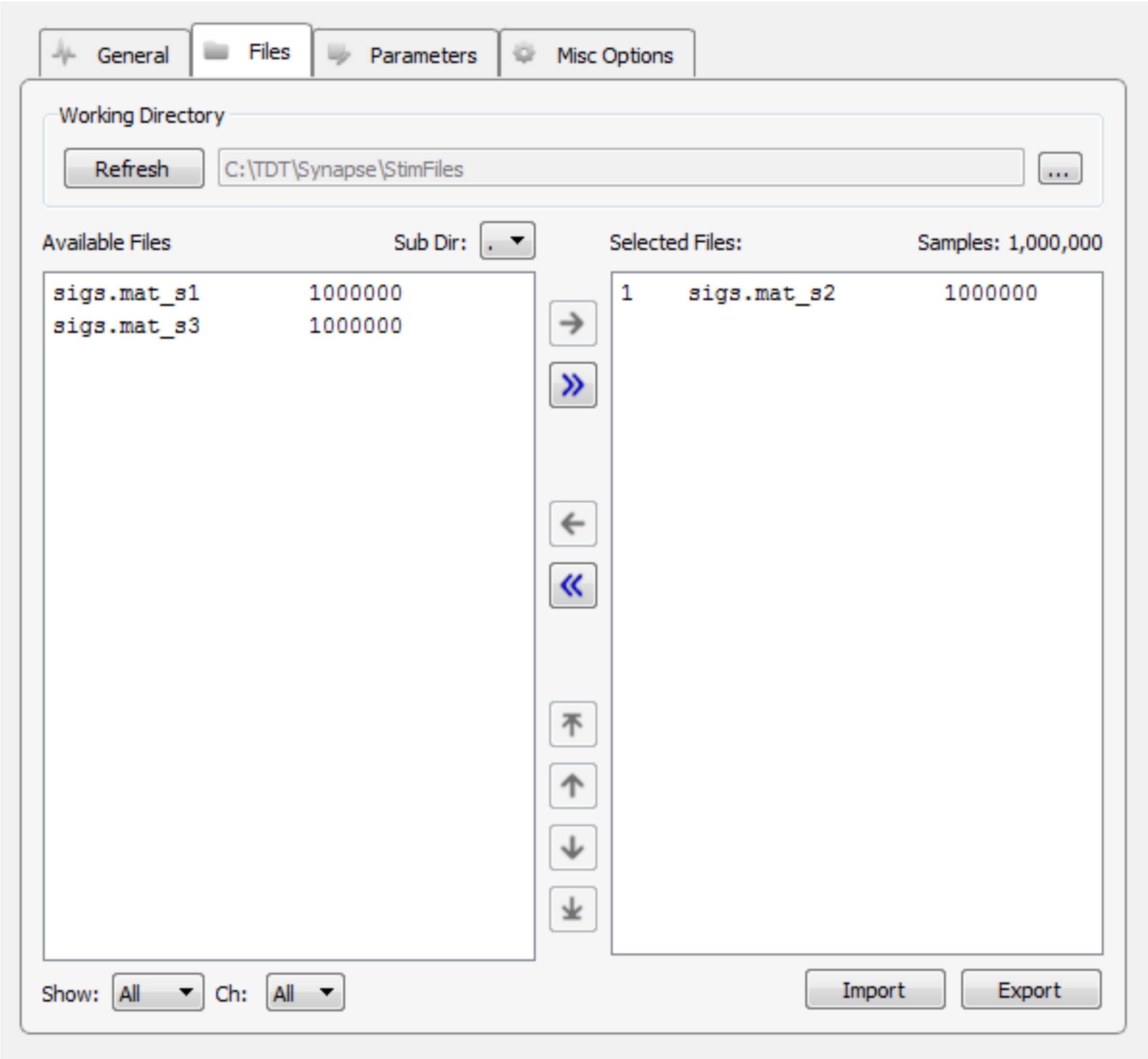
Signal Features

Features available here depend on the file type. When the check box is selected, the corresponding parameter is enabled in the parameter table.

Misc

In this section you can apply a signal gain factor and choose to output either a stimulus or pulse timing **Sync Output** signal.

Files Tab



Files Tab

Working Directory

The default working directory is C:\TDT\Synapse\StimFiles. You can select a different directory or stick with the default. If you add files to the directory or choose a new directory, you can click **Refresh** to update the displayed list of available files below.

The lower portion of the window serves as a simple graphical interface for displaying, filtering, and selecting stimulus files for play out.

Available Files

In the list on the left, all stimulus files found in the working directory are displayed. Stimulus files can be any of the following types:

- continuous 32-bit floating points (*.f32)

- continuous 32-bit integers (*.i32)
- continuous 16-bit integers (*.i16)
- Wave (*.wav)
- MATLAB arrays (*.mat)
- text file with one value per line (*.txt)
- text file with comma-separated values (*.csv)

A **Show Types** drop-down filter, below the **Available Files** area, narrows the displayed files to the selected file type. The **Sub Directory** drop-down menu allows you to drill down to subdirectories within the working directory.

Selected Files

The area to the right, serves as a list of files to be loaded as the stimuli.

File Buttons

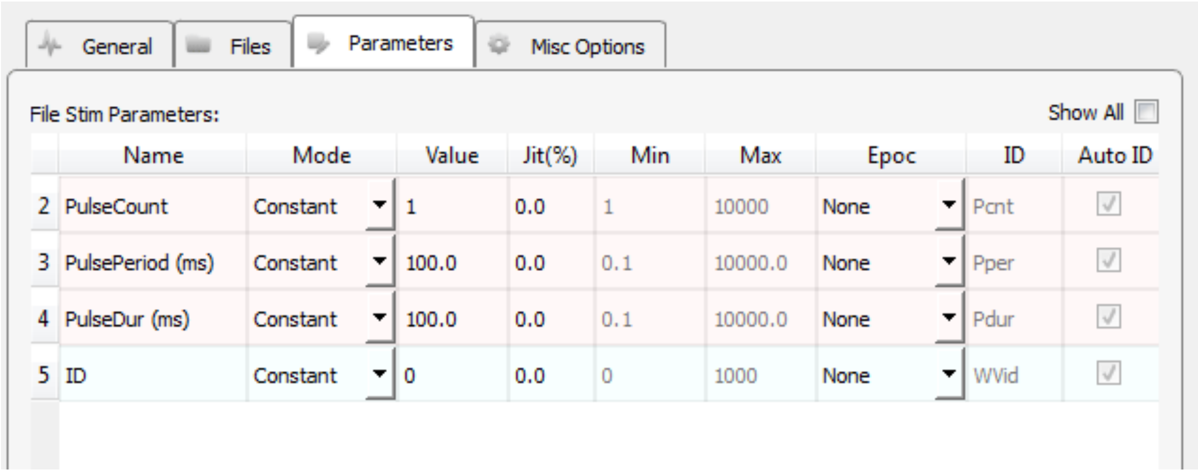
Use the file buttons, located between the two lists, to choose the files to use.

Button	Description
	move a file from available to selected
	move all files to Selected Files
	move a file from selected to available
	move all files to Available Files
	move file to top of list
	move file up in list
	move file down in list
	move file to the bottom of the list

Import /Export

These buttons can be used to import or export stimulus files.

Parameters Tab



Parameter Tab

File Stim Parameters

The table lists parameters relevant to configuring the stimulus. Each row represents a parameter and rows are shown or hidden in response to selections made on the General tab. Use the **Show All** check box to display hidden rows.

> **Tip**
>
> See Using Parameters for more information on controlling parameter tables.

Mode

In the Mode column, you can choose to make individual parameter Constant, controlled by a runtime Widget, or controlled by an Input line.

Value Columns

Enter values in the Value, Jit% (Jitter), Min, and Max columns to set the Constant value or to set the initial value when a Widget control will be used.

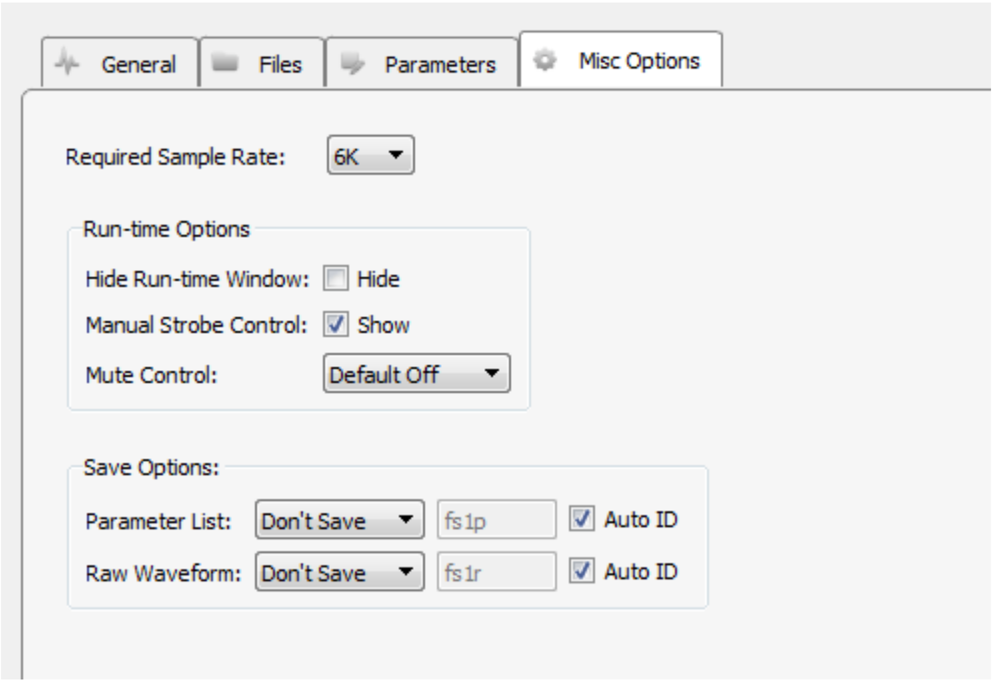
Epoc

In the Epoc column, you can choose to save the individual parameter value on stimulus or pulse onset.

ID and Auto ID check box

Synapse automatically generates a store name. TDT recommends using Auto ID to ensure no store names are duplicated. A "/" is appended to the name to indicate when the full epoc is stored (and is not appended when only saving the onset). To make your own store names, clear the **Auto ID** check box.

Misc Options Tab



Misc Options Tab

Required Sample Rate

The minimum rate required. Synapse looks through the entire system and sets the sample rate according to this and other limiting factors.

Hide Run-Time Windows check box

By default a runtime tab is added in preview or record mode. The contents of the tab are defined with configuration options on the General and Parameter options tab. Select the check box to hide the runtime tab.

Manual Strobe Control check box

When selected a manual strobe control is added to the runtime eStim tab. Clear the check box to hide the manual strobe control at runtime.

Mute Control

Select the default behavior of the runtime mute control. Mute allows you to mute or temporarily zero the stimulus during runtime. You can choose to hide or show the control and, if show, set the default start state.

Parameter List

Select whether to store the value of all parameters, at each stimulus or pulse onset. This generates a multi-channel list of scalar values. The channels map directly to the rows of the parameter table on the parameters tab. By default, some parameters are hidden in the table, but values for are stored for all parameters.

Raw Waveform

Select whether to store a copy of the raw stimulus waveform. You can choose to store continuously or only when the stimulus is active.

General Purpose Filter

Common Use Cases

The General Purpose Filter gizmo implements highpass, lowpass, and notch filters and supports control of corner frequencies at runtime.

Outputs	
Main	Single or multi-channel floating point filtered signal

Gizmo Help Slides

Quick Help Slide 1

General Purpose Filter — Create filters with user-defined parameters that include high/ low pass corners up to 8th order and notches with varying cut depths and bandwidths

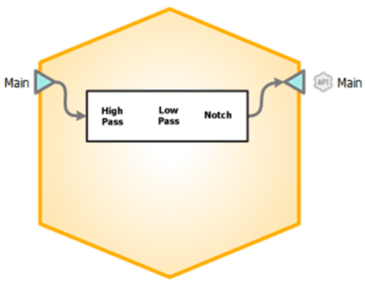
Common mode noise rejection

Customizable neural filters

Signal pre-conditioning

Multi-channel data stream from PZ5 being sent through filter before storage

Main: Input fed from neural amplifier or other floating point data source



Main: Output filtered signal to storage or other gizmos for further processing or visualization

Quick Help Slide 2

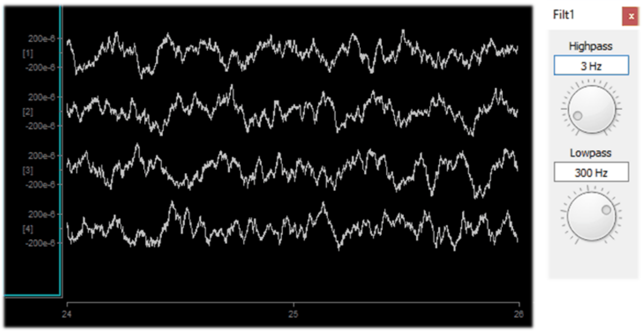
General Purpose Filter

User Interface, API, Data Stored

Dynamic control of Highpass and Lowpass filter settings

HighPassFreq — Control of high-pass frequency in Hz {Read/Write}

LowPassFreq — Control of low-pass frequency in Hz {Read/Write}



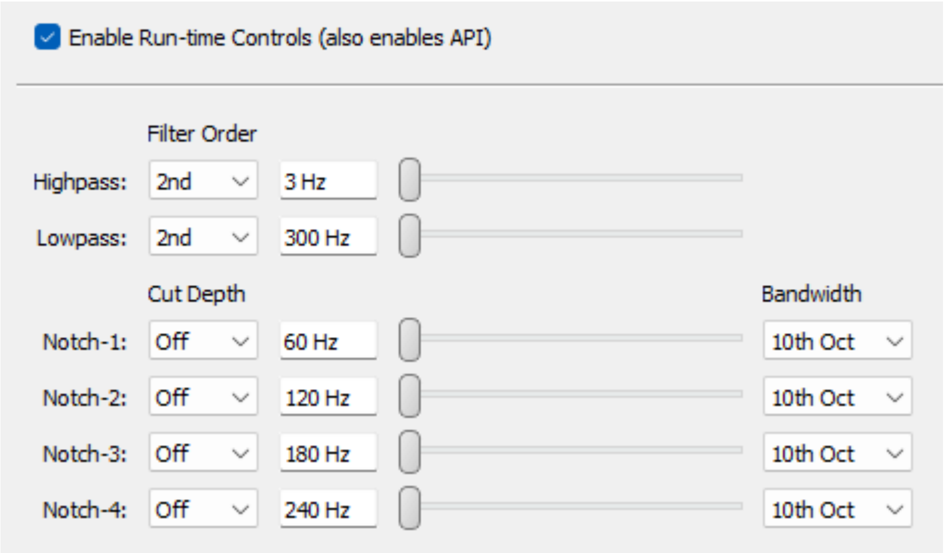
None

Reference

Filter Configuration Options

Set filter order, bandwidth, and default corner frequencies.

Options Tab



Filtering Options

When **Enable Run Time Controls Check Box** is checked, the corner frequency controls are added as a tabbed page in the runtime plot window.

> **Important**
>
> Notch filters can't be enabled/disabled at runtime. See the Neural Stream Processor gizmo if you require that.

> **Note**
>
> Filter settings are arranged with columns for settings and a row for each filter.

Filter Order is the number of bi-quad filters to use for the highpass/lowpass filters. Set the filter **Corner Frequency** for each filter.

Cut Depth is how effective each notch filter is (in dB). Set the notch **Center Frequency** and **Bandwidth** (in octaves) to determine the sharpness of the notch filter.

Injector

Common Use Cases

Insert a single channel input into a multi-channel data stream at specific user-specified channels. Choose a channel for electrical stimulation. Can also be used to route audio signal to a speaker array (channel in DAC Montage).

Data Stored	
Epoc (optional)	User selected channels, when any channel changes
Outputs	
Output	Multi-channel floating point output
ChanSel-* (optional)	Selected channel numbers
Par Output	Full parameter stream

Gizmo Help Slides

Quick Help Slide 1

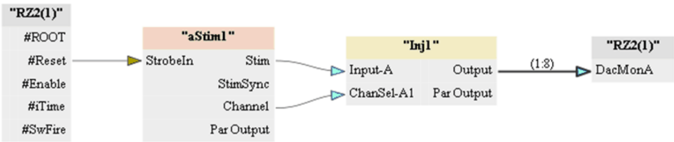
Injector — Insert a single channel input into a multi-channel data stream at specific user-specified channels

Route audio stimuli to specific speakers in an array

Route electrical stimuli to specific channels in an electrode array

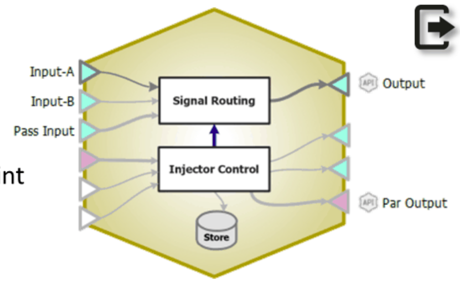
Convert a single channel stream into a multi channel stream (Injector configured as Generator)

Audio stimulation signal is injected into a specific speaker in an array. The Channel parameter in aStim was manually created.



Input-:* Signals to inject into a multi-channel stream

Pass Input: Multi-channel floating point signals, good for cascading multiple Injectors



Output: Output the multi-channel signal

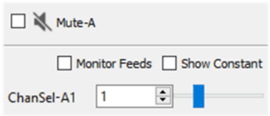
Par Output: Contains info about currently selected channels

Quick Help Slide 2

Injector

User Interface, API, Data Stored

Mute injected channels and dynamically control injector channel.



*ChanSel-** — Pick the channel inject the signal input onto {Read/Write}

*Mute-** — Mute the selected channel {Read/Write}

None

Quick Help Slide 3

Injector

Parameter Table

This gizmo has a Parameter Table to dynamically control values its parameters from other gizmos and during runtime

Parameter Tables provide the user dynamic control over parameter values. Each one of the Modes will change how the user can control and interact with the parameter.

Param In – Dynamically control parameter values from Parameter Sequencer or Parameter Manifold

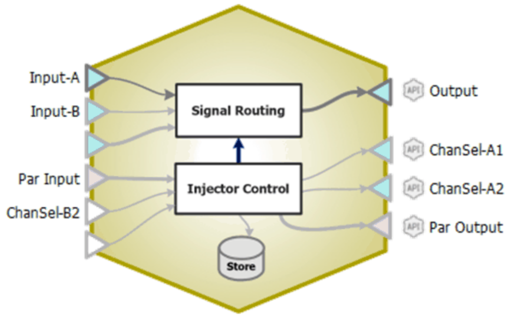
Constant – The value is immutable at runtime

Scalar In – Pass a scalar value from another gizmo for dynamic real-time control

Widget – Creates a widget (slider) on the user interface that is controlled manually by the user or through the API.

Scalar Out (Scout) – Makes this value available as a gizmo output to connect to another gizmo input in real-time

Epoc – Save the parameter values used for each stimulus presentation



Changing the Mode of parameters will change the input and output options on the gizmo.

All parameters can be read through API. Only parameters in Widget mode can be written to.

All parameters are connected to the "Par Output" gizmo output link

General | Parameters

Signal Injector Parameters: Show All ☐

	Name	Mode	Value	Min	Max	Epoc	ID	Auto ID	SCout-1	SCout-2
3	ChanSel-A1	Param In	0	0	16	None	ChA1	☑	◉	○
4	ChanSel-A2	Param In	0	0	16	None	ChA2	☑	○	◉
7	ChanSel-B1	Widget	0	0	16	None	ChB1	☑	○	○
8	ChanSel-B2	Scalar In-1	0	0	16	None	ChB2	☑	○	○

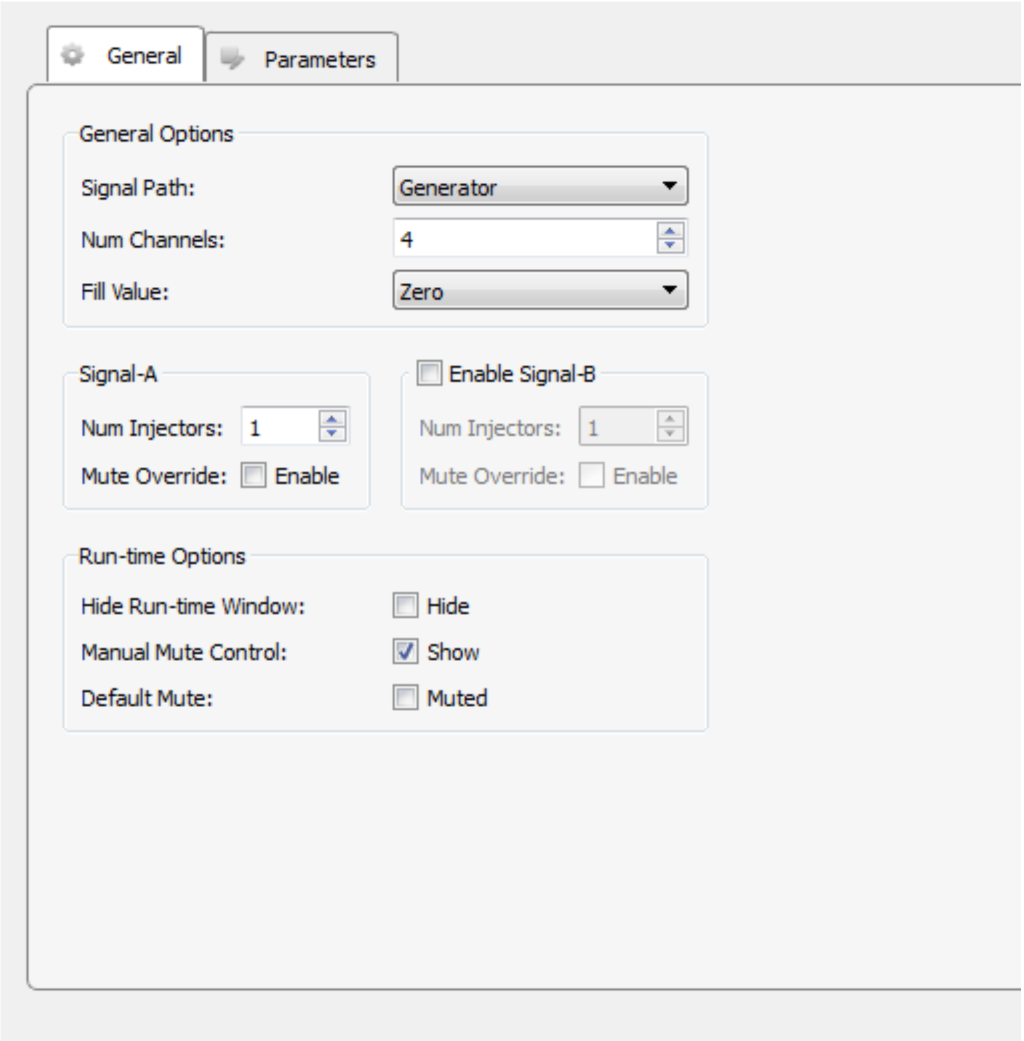
Reference

The Injector inserts a single channel input into a multi-channel data stream at channels you specify. For example, use this to send a given stimulation pattern to one or more specific channels on a stimulator, with full dynamic control at runtime.

> **Important**
>
> For more advanced electrical stimulation control, see the Electrical Stim Driver gizmo

Injector Configuration Options

General Tab



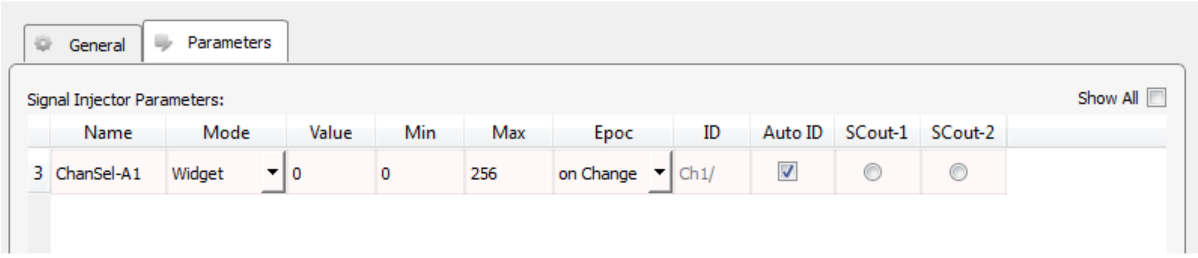
General Tab

General Options

The Injector can operate on an existing multi-channel stream (**Pass Thru**) or generate a multi-channel stream of constants (**Generator**) with the given **Fill Value**.

Each single channel input (**Signal-A** and **Signal-B**) can be injected on up to four channels, chosen in the **Parameters** tab.

Parameters Tab



Parameter Files Tab

Use the parameters table to define how the injection channels are selected. See Using Parameters for more information on working with parameters tables.

Local Field Potentials (LFP)

The Local Field Potentials (LFP) gizmo filters multi-channel waveforms to display and record LFP activity.

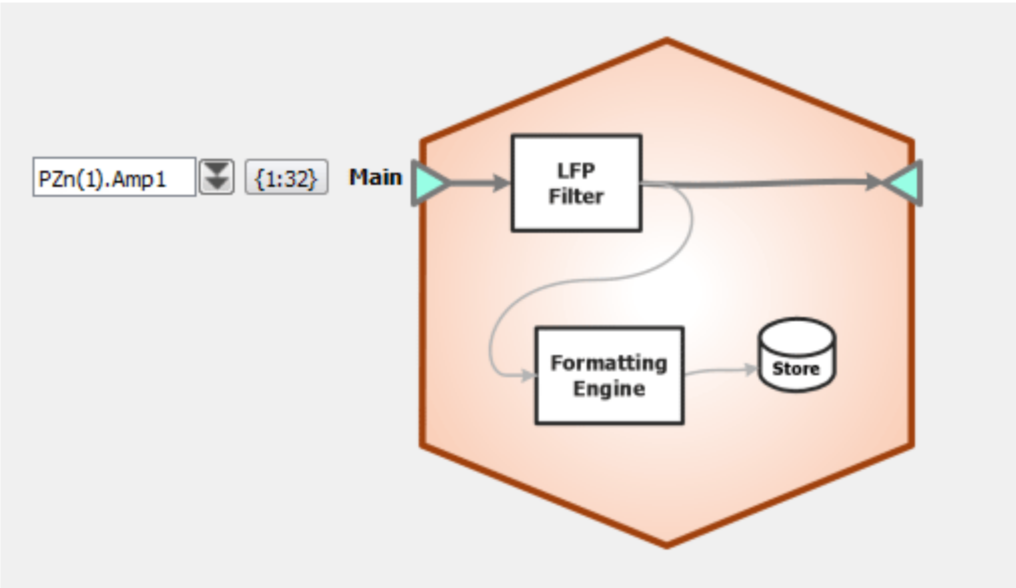
Important

The LFP gizmo was replaced by the Neural Stream Processor gizmo. It is only available if you enable Deprecated gizmo in Menu → Preferences.

Data Stored	
Stream (optional)	Continuous filtered waveforms
Outputs	
Main (optional)	Filtered waveforms

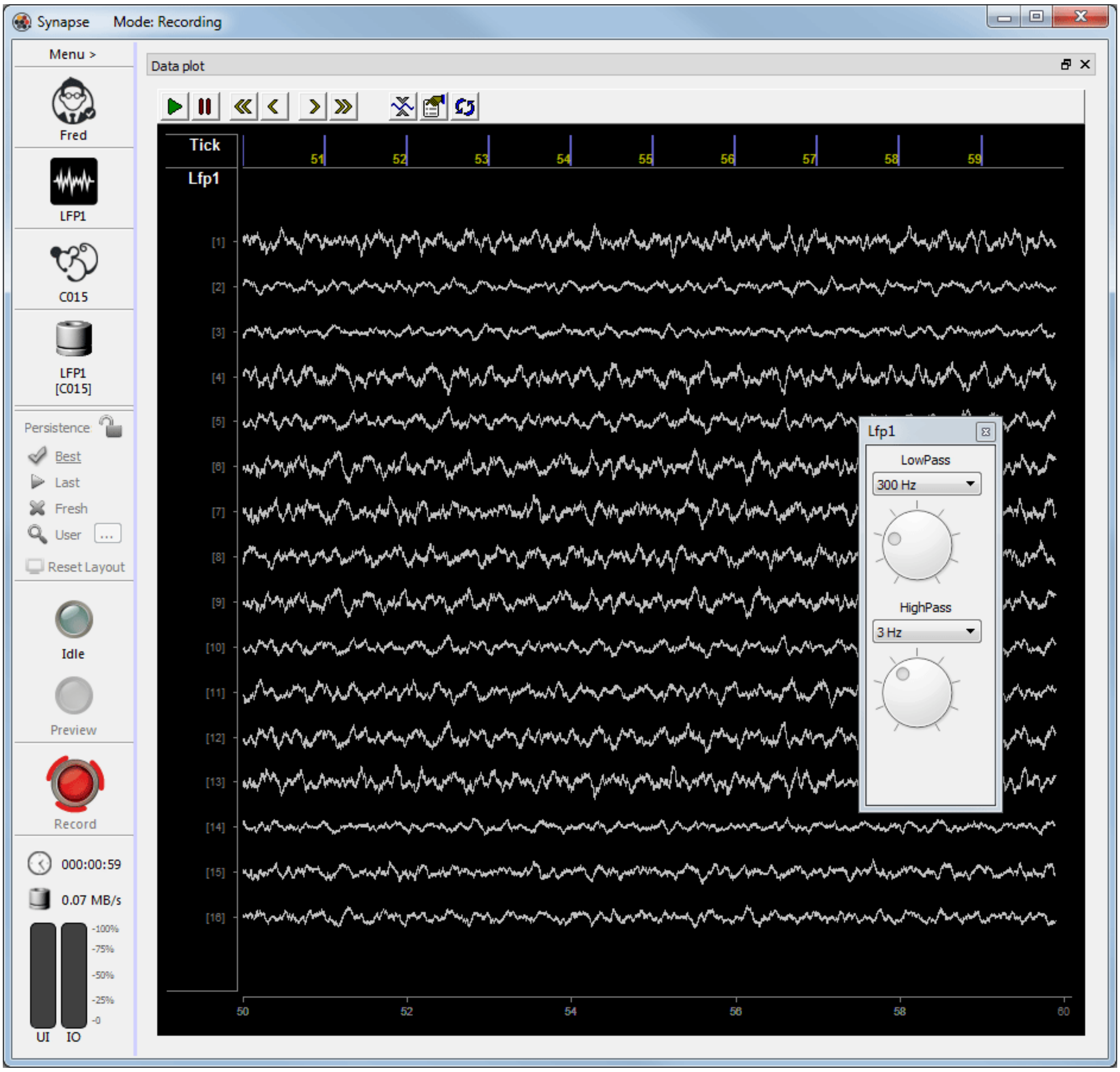
Reference

The LFP gizmo takes multi-channel floating point signals, filters the signals and optionally formats and stores into the data tank. The filtered data can also be available as an output to other gizmos for further processing.



LFP Filter Block Diagram

The LFP Runtime Interface



Runtime Window

Runtime Plot

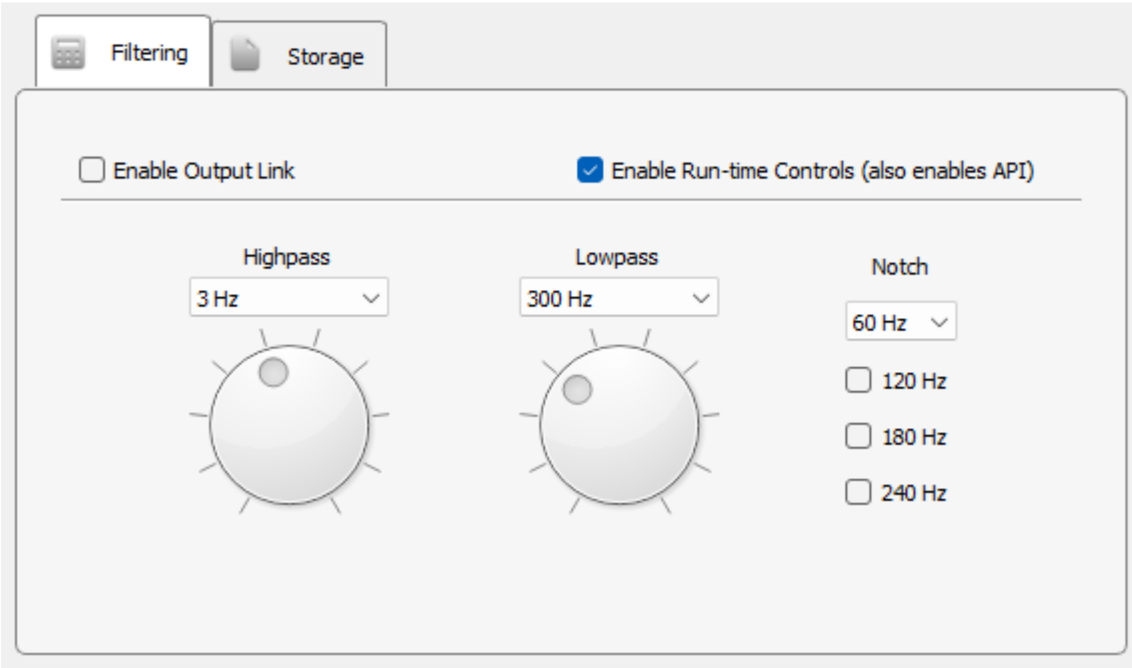
A multichannel streaming plot is included in the data plot tab when storage is enabled. See Flow Plot for more information on using and customizing the plot.

LFP Tab

The LFP tab contains controls for runtime highpass and lowpass filter adjustments, if the **Enable Run Time Controls** option is selected at designtime.

LFP Configuration Options

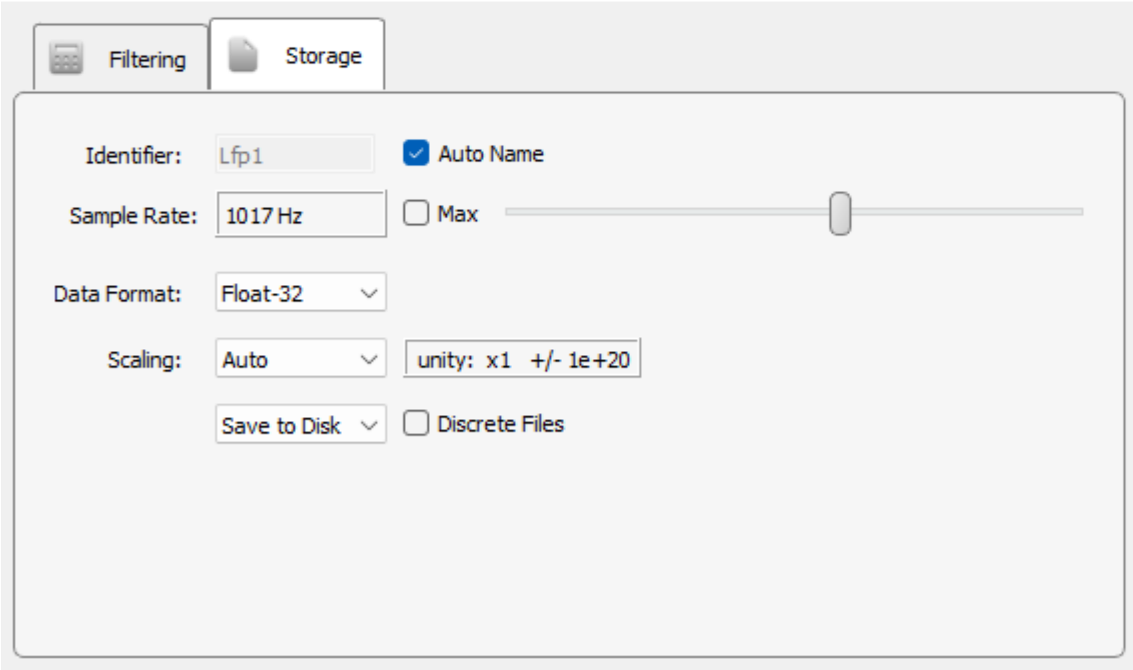
Filtering Tab



Filtering Options Tab

Select the initial highpass, lowpass, and notch filter values. To modify the highpass and lowpass filter settings during runtime, select the **Enable Run Time Controls** check box. Use the **Enable Output Link** check box to make filtered waveforms available as an output from this gizmo.

Storage Tab



Storage Options Tab

Set the name, data format, scaling factor, and sampling rate of the stored data. Drag the slider until the desired rate is displayed.

Use the **Discrete Files** check box to save each channel of data as a discrete file (*.sev file) in the data tank.

Clear the **Save to Disk** check box to view data in the runtime plots without storing data to the Tank.

Mapper

Common Use Cases

Create user-defined channel maps to reroute electrode sites to different amplifier channels. Use this when you want to create ordered spatial maps with unordered electrode sites.

Outputs	
Stream	Remapped multi-channel waveforms

Gizmo Help Slides

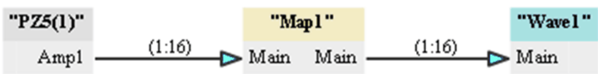
Quick Help Slide 1

Mapper — Create user-defined channel maps to reroute electrode sites to different amplifier channels

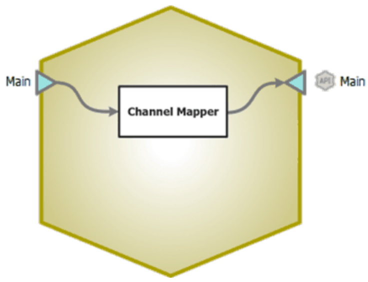
Silicon probe recordings

Electrode Interface Board (EIB) recordings for microdrives

16-Channel output from PZ5 sent to the Mapper for reordering before storage



Main: Input fed from neural amplifier



Main: Output the reordered channels for storage, sorting, or filtering

Quick Help Slide 2

Mapper

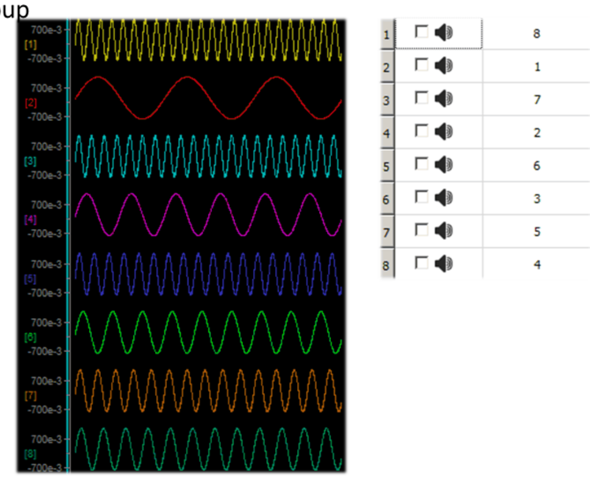
User Interface, API, Data Stored

Mute all or select channels in mapping group to zero value

ChanMap — Array of values corresponding to the channel map list {Read/Write}

MuteMap — Mute all the channels in the map {Read/Write}

Custom Map — Save the custom map made in Design Time to a csv file



Each channel in the pre-mapped signal corresponded to a sine wave with a frequency of that channel number.

Reference

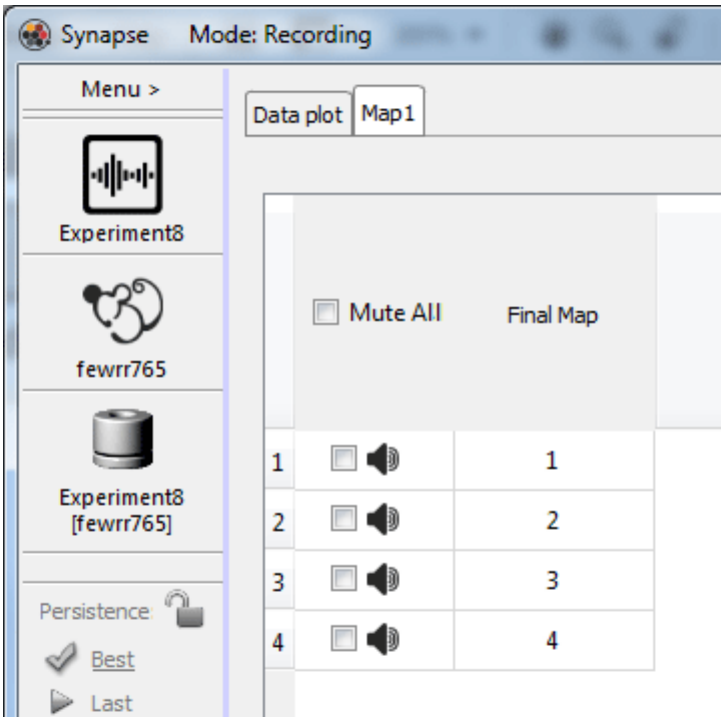
Mapper provides a simplified interface for remapping recording channels. It takes in multi-channel signals then remaps or reorganizes the channel order for your system. You select your electrodes, headstage, and adapters from lists or edit the map manually for custom system components.

See the full Channel Mapping Guide here.

The Mapper Runtime Interface

Map Tab

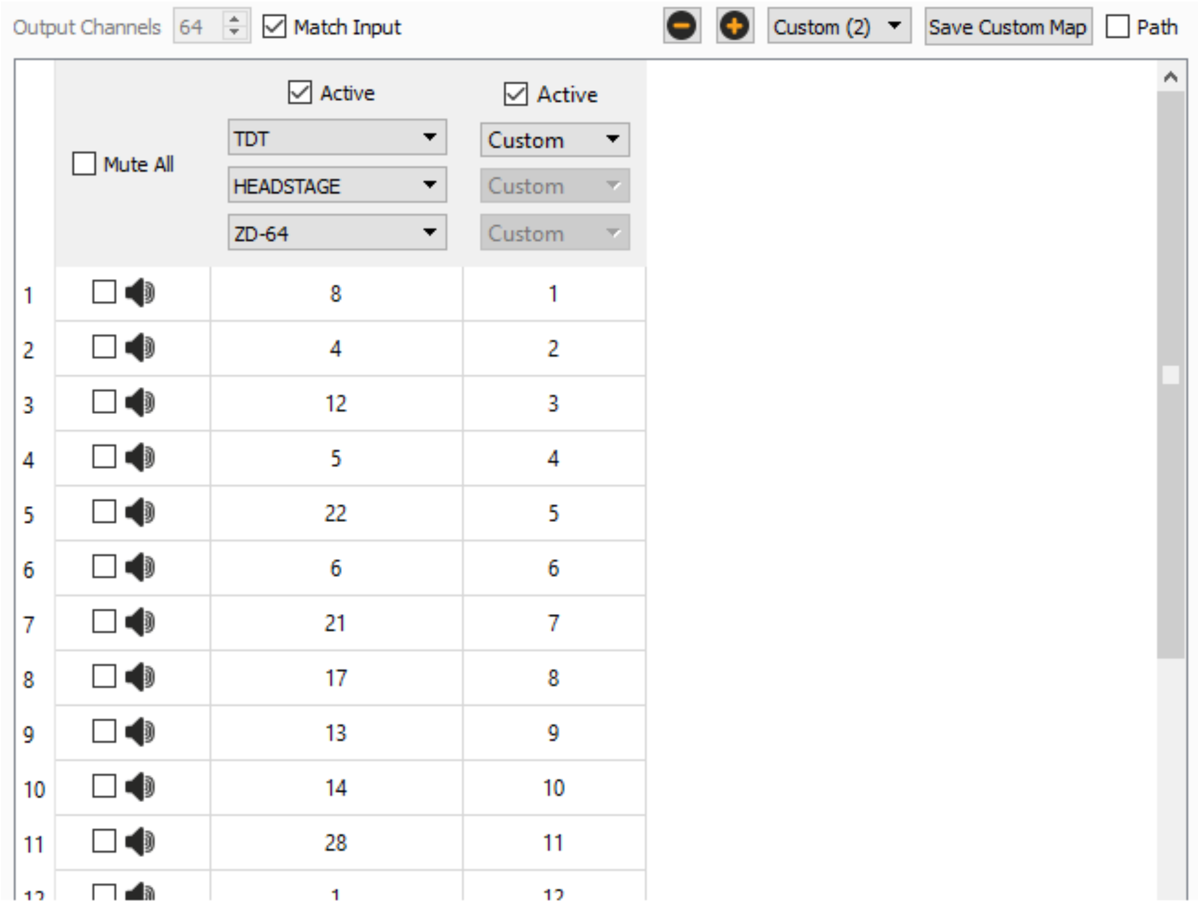
At runtime the channel maps table is displayed for viewing the final map and allowing you to mute noisy channels.



Runtime Tab

Mapper Configuration Options

Options Tab



Mapper Options

Use the drop-down menu to choose an existing map for your Headstage, Adapter, or Electrode, or create your own Custom map. The default maps are read from a CSV file that installs with Synapse (C:\Synapse\SupportFiles\EAHS.csv). You can add your own maps to this CSV file and they will appear in the drop-down list.

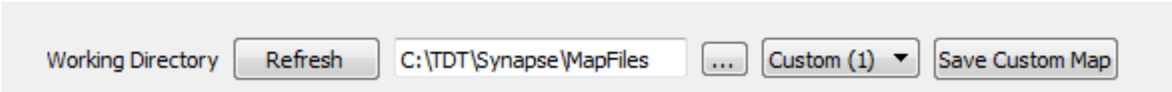
You can enter the channel map manually, or you can copy it from the clipboard by right-clicking on the starting channel that you want to paste the map into.

To only pass a subset of channels through the Mapper gizmo, clear **Match Input** and change the number of **Output Channels**.

Click the "**-**" icon to delete the selected column. Click the "**+**" icon to add a column to the map. The new column is added to the right of existing columns. All active maps will be applied to the incoming data stream. The **Active** check box must be selected to allow editing.

Use the **Mute** check boxes to set the default mute state of each channel.

Working Directory

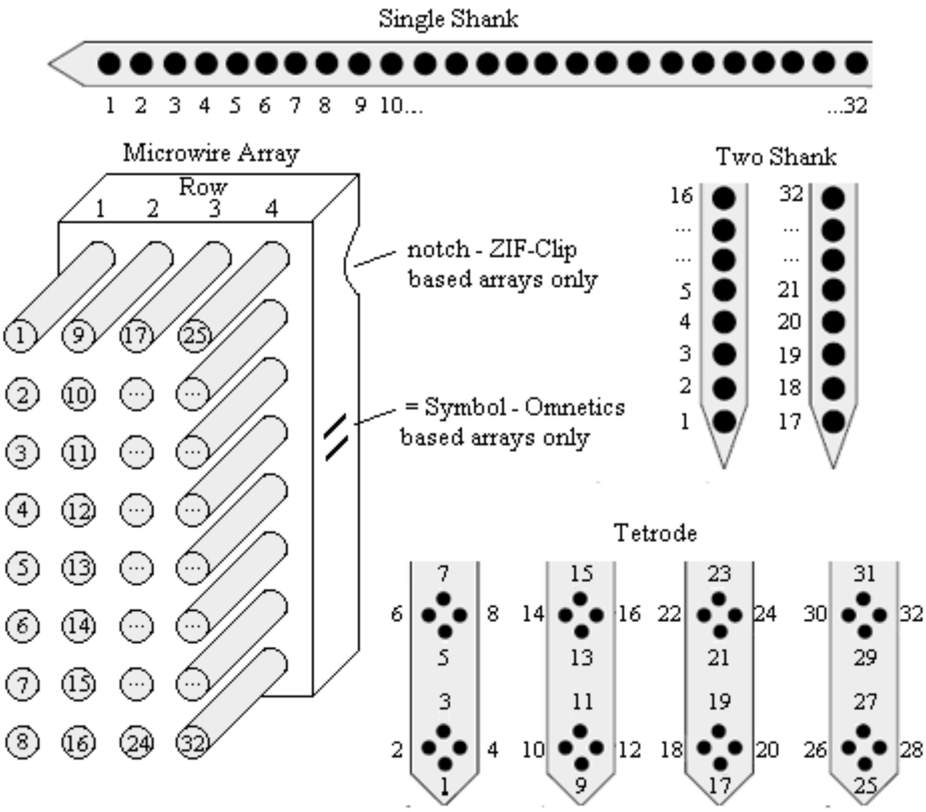


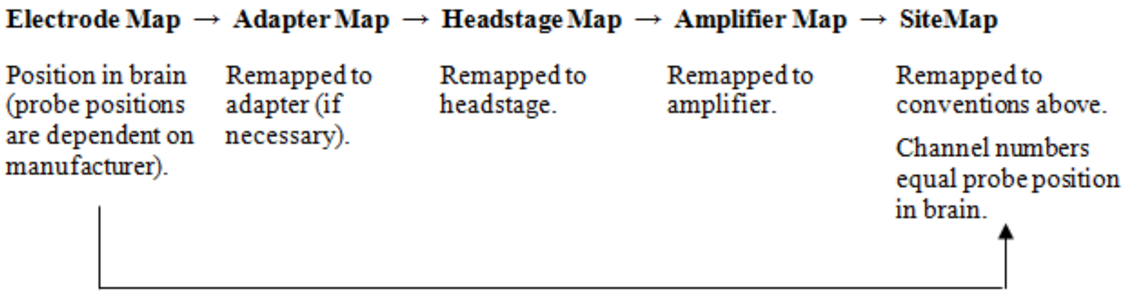
Working Directory Options

Directly above the mapping interface, you can save the map as a Custom Map or open an existing Custom Map.

Site Numbering Conventions

Probe sites for shanks and tetrodes are arranged clockwise and in ascending order from tip-to-shank. Omnetics and ZIF-based microwire arrays are arranged in descending order top-to-bottom from left-to-right with the array symbols shown in the diagram below.





Site Numbering Conventions

Merger

Common Use Cases

Combine up to eight single or multi-channel streams into a single multi-channel stream. Use this gizmo to send separate data into a single multi-channel stream for processing in other gizmos or storage.

Outputs	
Main	Merged multi-channel signal

Gizmo Help Slides

Quick Help Slide 1

Merger — Combine up to eight single or multi-channel streams into a single multi-channel stream

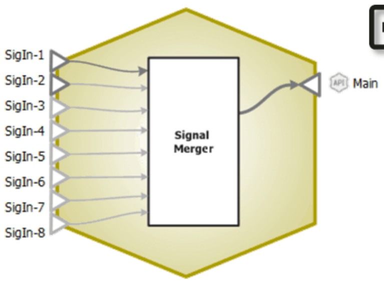
High channel count neurophysiology

Send individual LFP and EEG channels to a custom sleep state detector

Multi-channel signals from separate logical amplifiers are sent to a Merger to be stored onto an RS4

SigIn: Single of multi channel streams. Int and TTL can be mixed. Floats can't be mixed with other data types.

Once a stream is declared (single or multi) then only that type can be used.



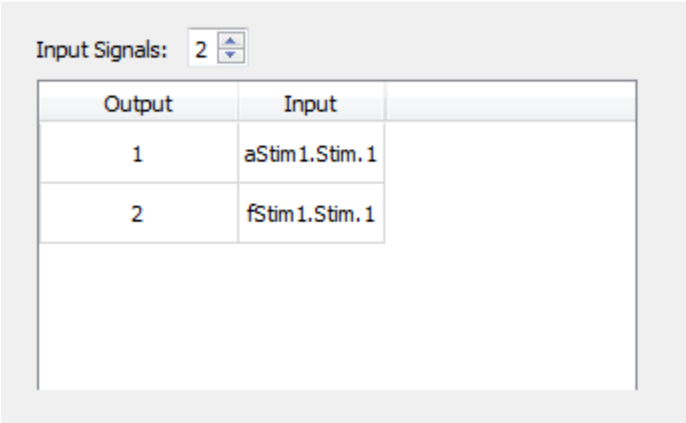
Main: Output the combined signals as a multi-channel stream

Reference

The Merger gizmo takes up to eight single-channel integer or floating point inputs, or up to eight multichannel inputs, from other gizmos or HALs and merges them into a single multichannel output.

Merger Configuration Options

Options Area



Merger Configuration Options

Each Merger gizmo can take either single channel or multi-channel inputs, but can't mix them. The source that is first assigned to the SigIn-1 input determines which signal type can be selected for subsequent sources. The type can't be changed without deleting the gizmo.

Input Signals value box

Select the number of input signals that you want to merge into one multi-channel output (between 2 and 8).

When increasing the number of signals, commit the change then display the block diagram to select the additional input sources. Commit again to see them updated in the matrix.

The output channel count is always a multiple of two, and is always greater than or equal to four.

MRI Recording Processor

Common Use Cases

Suppress MRI recording artifacts using controllable signal gate. Titrate gating tightly around artifact to clean up online signals. Use this gizmo to eliminate gradient switching artifact in an MRI recording environment. Can automatically detect artifacts or be triggered using timing signal from the magnet.

Data Stored	
Epoc (optional)	Timestamps for each artifact
Outputs	
Continuous (optional)	Single Unit filtered output
Continuous (optional)	LFP filtered output

Gizmo Help Slides

Quick Help Slide 1

MRI Recording Processor — Suppress MRI recording artifacts using controllable signal gate. Titrate gating tightly around artifact to clean up online signals

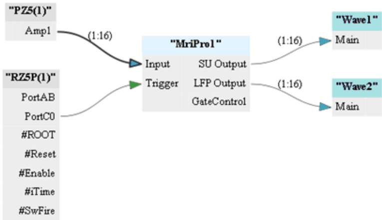
Recording inside an MRI

Neural recording with electrical stimulation

Startle response suppression

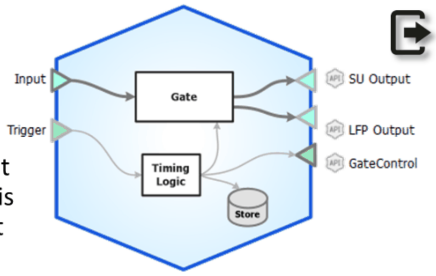
Transducer input processing

Gradient switching artifact is suppressed using external timing signal from magnet, then output to storage



Input: Input fed from neural amplifier or other high sensitivity signal

Trigger: Optional input from logic signal that is time-locked to artifact



SU Output: Single unit data passed to storage or visualization gizmo

LFP Output: LFP data passed to storage or visualization gizmo

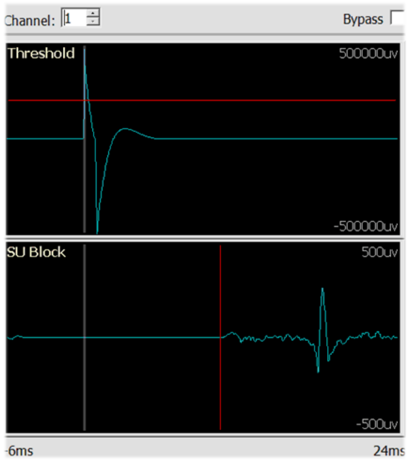
GateControl: Time-locked logic signal reflecting gate control

Quick Help Slide 2

MRI Recording Processor

Gizmo Configuration, Data Stored

Dynamic control of artifact threshold and gate width to titrate signal blocking



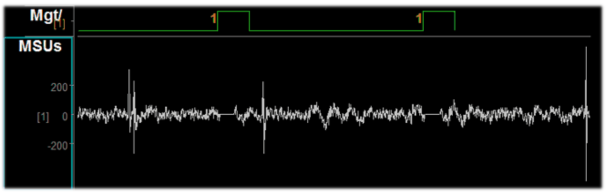
Gate Control — Automatic artifact detection or trigger from an external TTL

Working Window — Duration of runtime snippet. Also determines the minimum time between artifacts

Center at — Centers the runtime window around the artifact onset

Mgt/ — Gate Timing signal is a full epoch capturing gate onset and window offset {Epoch}

Mlfp/MSUs — Filtered LFP and Single Unit streams with artifact removed {Stream}



Reference

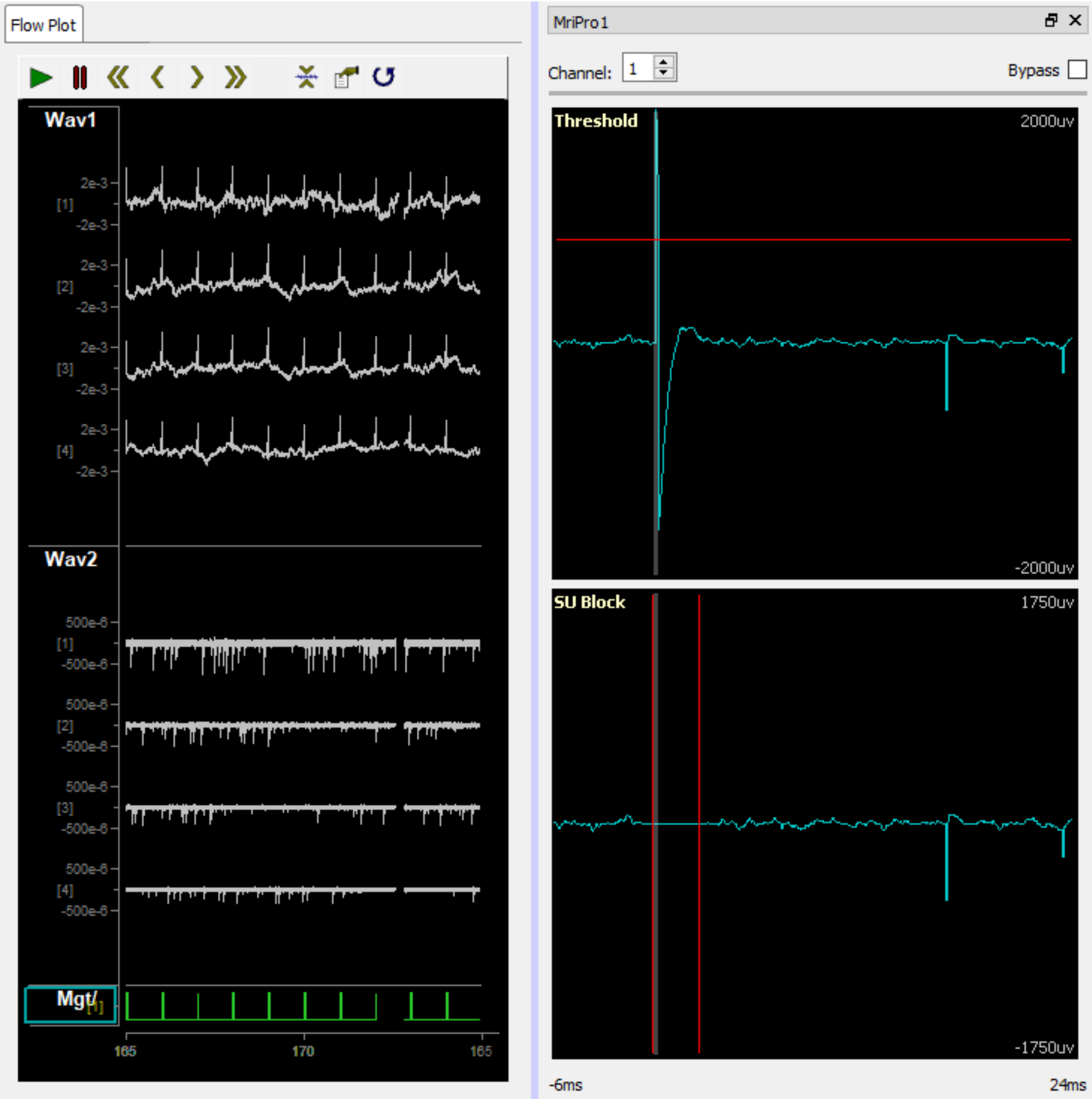
The MRI Recording Processor gizmo removes MRI scanner artifacts from Single Unit and LFP data in real time. It can automatically detect/reject artifacts in the data stream or be synchronized with an external TTL. Timing logic can also be stored.

Typically the input is the raw amplifier signals and the output Single Unit data is ready for online spike sorting gizmos.

The MRI Recording Processor Runtime Interface

Runtime Plot

If you choose to save gate timing, a plot showing the timing of the gate is added to the runtime window for visualization.



Runtime Plots include MRI Artifact Timing

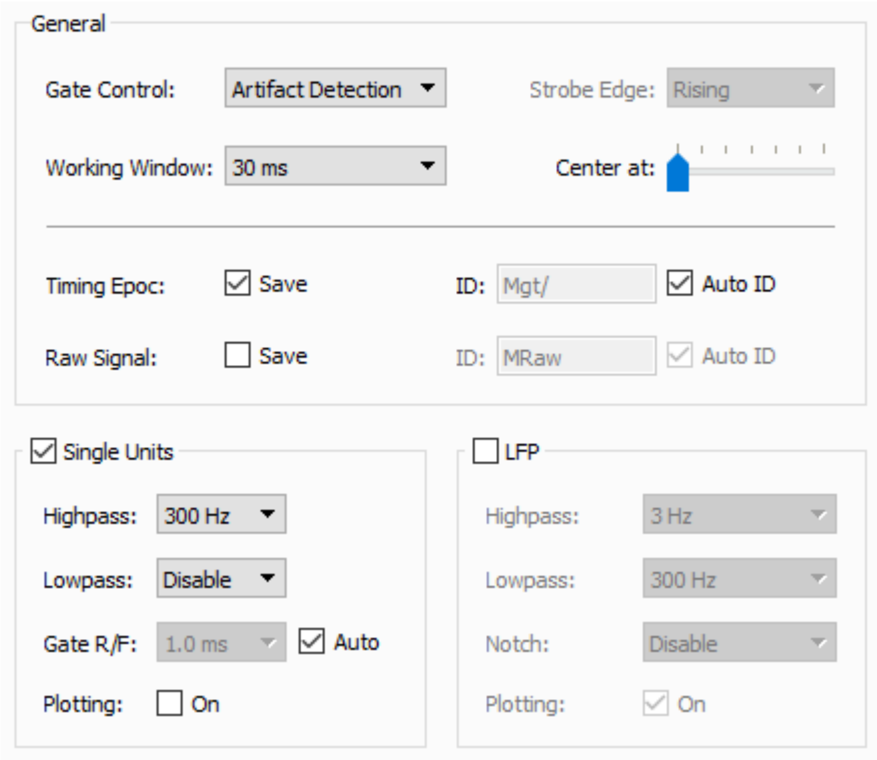
The threshold plot lets you choose a threshold for automatic artifact detection/rejection. To adjust the y-scale, use the **mouse wheel** or **Shift + Left-click drag** the mouse. Right-click the choose **Find Threshold** to determine a reasonable threshold before fine tuning.

The lower plots (SU Block and LFP Block, if enabled) let you set a blanking window around the artifact onset. Adjust the left and right vertical bars to create as small a blanking window as possible while still removing the artifact. Click the **Bypass** checkbox to see what the waveform looks like without the rejection applied.

In the Flow Plot above, Wav1 is the raw signal going into the blocker with a >2 mV artifact, and Wav2 is the output of the single unit data with artifacts removed so you can see the spike waveforms are still present but without the large artifact.

The timing signal of the rejection window is shown at the bottom in green.

MRI Recording Processor Configuration Options



Trigger and Storage Options

General

The **Gate Control** can trigger off a user-defined threshold crossing at runtime. If the artifact is timed to another gizmo or an external TTL event, those can be used as the gate trigger instead.

Working Window determines what size of snippet to show in the runtime plots for gate control, and also determines the minimum time between consecutive artifact detections. If you find that multiple artifacts occur in the same window, consider reducing the working window size.

Center at determines where to position the artifact onset in the runtime plots.

> **Important**
>
> The MRI processor operates on a delayed version of the waveform so it can effectively remove the artifact. The *Center At* slider ranges from 20% to 80% of the working window, so with *Center At* at the far left slider position you should see ~20 ms delay when *Working Window* is 100 ms.
>
> Setting the Working Window smaller will decrease your signal delay, but this will only work if your artifact is shorter than the *Working Window*.

Timing Epoc stores the rejection timing signal in the data tank.

Raw Signal stores the raw streaming data before the rejection is applied.

Single Units

Set the filter characteristics for the Single Units signals before artifact rejection.

The Single Unit artifact blocker uses a cosine-squared gate. The rise/fall time (**Gate R/F**) represents the amount of time it takes to reduce the signal by 90%, and to increase it back to 90% of its final value. When set to **Auto** it automatically adjusts based on the **Working Window** size.

Plotting adds the resulting stream to the Flow Plot.

LFP

Set the filter characteristics for the LFP signals before artifact rejection.

The LFP artifact blocker uses a custom rejection method to reduce effects after gating.

Plotting adds the resulting stream to the Flow Plot.

Neural Signal Referencer

Common Use Cases

Digitally subtract common signals from multi-channel stream. Single or multi-channel referencing on all channels or independent sub-groups of channels. Use this gizmo to eliminate common mode noise across channels or to perform digital re-referencing. Multi-channel referencing won't create artificial waveforms on your signal.

Data Stored	
Stream (optional)	Continuous reference waveforms
Outputs	
Main	Re-referenced multi-channel floating point signal
RefOut	Single-channel computed reference signal

Gizmo Help Slides

Quick Help Slide 1

Neural Signal Referencer — Digitally subtract common-averaged signals from multi-channel streams. Perform single or multi-channel referencing on all channels or independent sub-groups of channels

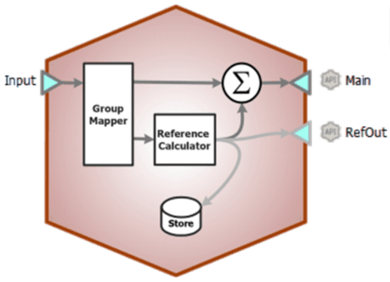
Common mode noise removal

Digital re-referencing

Freely-behaving recording experiment

16-Channel data stream sent from PZ5 to LFP filter, then to Neural Referencer for digital re-referencing and storage

Input: Input fed from neural amplifier or channel mapper



Main: Output the re-referenced data stream. Signals are re-combined in one stream if multiple reference groups were used.

RefOut: Output the neural reference signal. This is also stored internally in the gizmo.

Quick Help Slide 2

Neural Signal Referencer

Gizmo Configuration, Data Stored

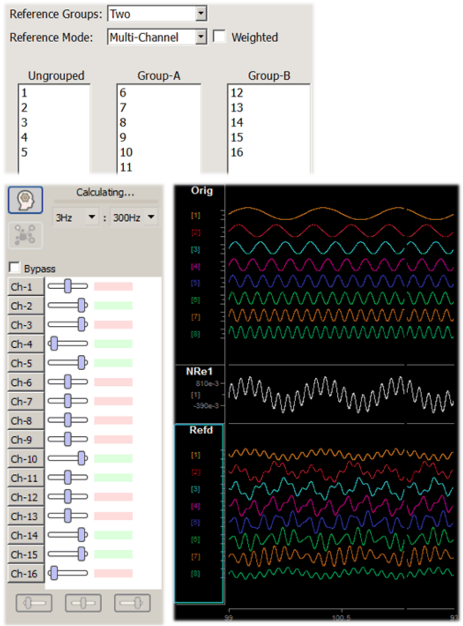
Select channels to use or exclude in reference groups in multi-channel Reference Mode. Determine channel correlation within a group using cluster analysis.

Reference Group — Select between Simple Single or multiple Reference Groups.

Moving the slider to the right includes the channel in the calculation and applies the re-referencing.

Moving the slider to the left excludes the channel and does not apply the re-referencing.

NRe1 — The reference signal {Stream}



Channel 1 (1Hz) and 8 (8Hz) of "Orig" were used as reference signals

Reference

The Neural Signal Referencer gizmo takes multi-channel floating point signals, determines the common signal on all or independent sub-groups of channels, and removes it. The signals can optionally be normalized before the reference is calculated. The resulting signals (and optionally the reference signal used for the subtraction) is available as an output to other gizmos for further processing.

The Neural Signal Referencer Runtime Interface

Runtime Plot

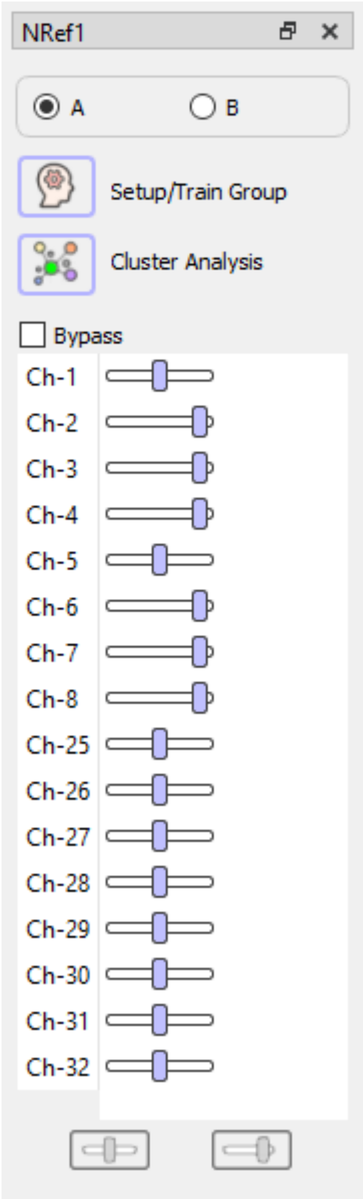
A streaming plot of the reference signal used for each group is optionally included in the data plot.

Neural Referencer Tab

The Neural Referencer tab contains controls for choosing the referencing channels for each group. In single-channel reference mode, a slider for each reference group chooses the channel to use to create the reference signal that is subtracted from that group.

Multi-Channel Mode

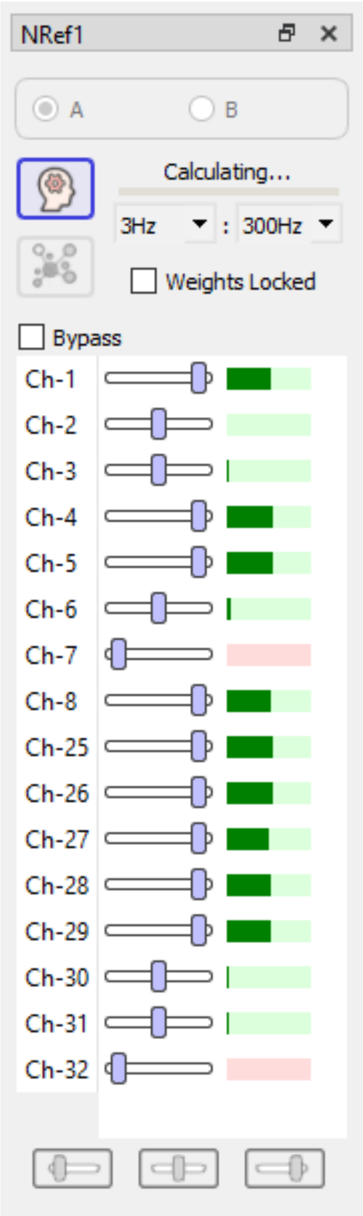
In multi-channel reference mode, there is a slider for each channel.



Runtime Tab without Weighting

If Weighting is disabled, the average of all channels in the group is used as the reference signal and the channel sliders have two positions. If the channel slider is in the 'center' position, the reference signal is not subtracted from that channel. If the channel slider is in the 'right' position, the reference signal is subtracted from that channel. Therefore, to subtract the average signal from all channels, move all sliders to the 'right' position.

With Weighting

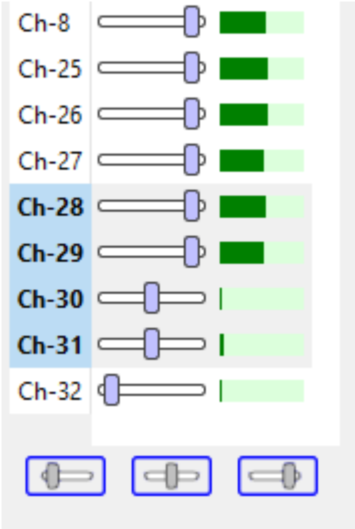


Runtime Tab with Weighting

If Weighting is enabled, you can selectively include/remove channels from the reference signal average. To do so, initiate training by clicking the 'Setup/Train Group' button. Training computes the correlation between each channel and the current reference signal and displays the result as a green (correlated) or red (uncorrelated) bar next to each channel. This helps you decide which channels to include (bigger green bar) or remove (bigger red bar, move channel slider to the 'left' position) from the reference signal. The included channel contribution to the reference signal is weighted based on the correlation factor.

Tip

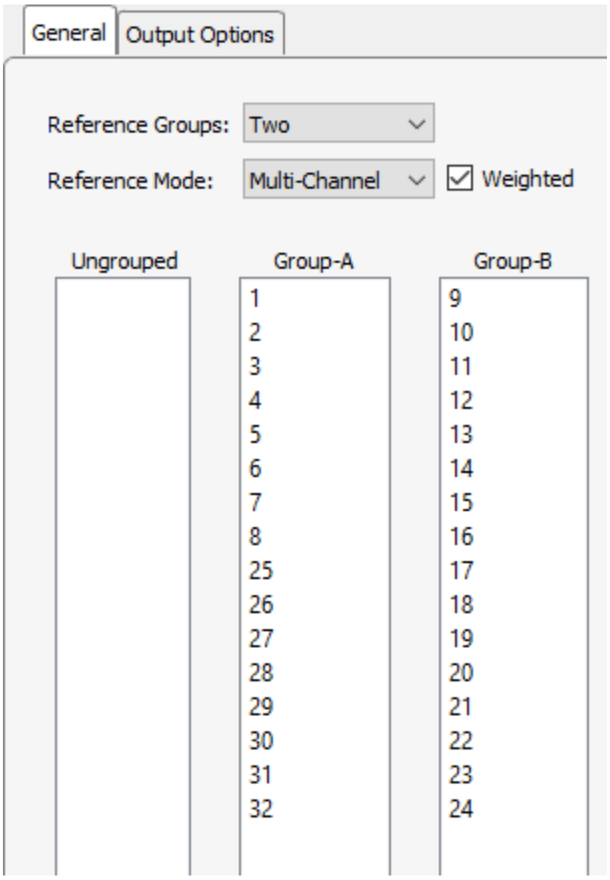
To change multiple sliders at once, click and drag the channel number labels or use the CTRL key to select multiple channels, then use the buttons at the bottom to change them all at once.



The Cluster Analysis mode will suggest channel groupings based on a minimum correlation threshold. This can help refine the groupings that were chosen at design time.

Neural Signal Referencer Configuration Options

General Tab



Neural Signal Referencer General Tab (Multi-Channel Mode Shown)

Choose up to four starting groups for your referencing. Each group will be referenced separately, and are then merged back together in the correct order.

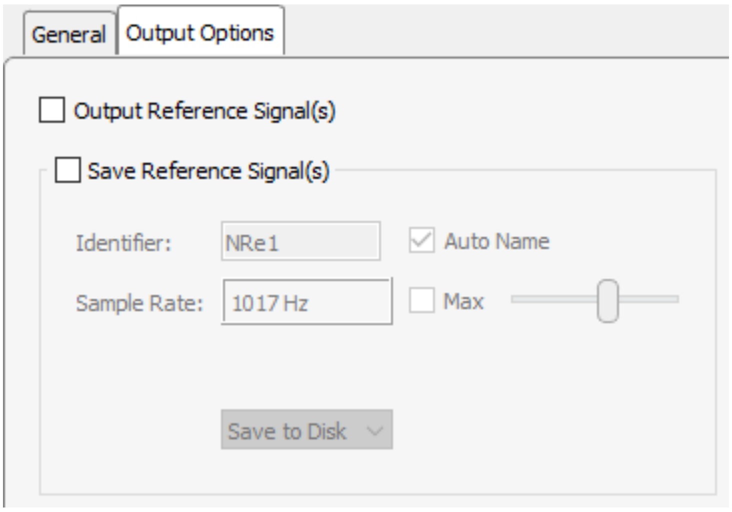
By default, in **Single Simple** mode all channels are treated as one group and you can pick a single channel as a digital reference that is subtracted from all channels.

In **Signal Channel** mode, each group can have a single channel from within the group act as a reference for that group.

In **Multi-Channel** mode, any number of channels can be added to the reference signal that is subtracted from the group.

In **Weighted** mode, the channels in each group are normalized before calculating the reference signal, and then each individual scaling factor that was used is removed when the reference signal is subtracted from each channel.

Output Options Tab



Output Options Tab

Select **Output Reference Signal(s)** to make the reference signal(s) available as gizmo output(s).

Select **Save Reference Signals(s)** to display and/or save the reference signal(s) to disk. The Identifier is used to name the data store that is saved in the tank. It must be four characters in length.

Choose a specific Sample Rate for the data store, or set it to Max and it will run at the master device rate.

Clear the **Save to Disk** check box to view data in the runtime plots without storing data to the Tank.

Neural Stream Processor

Common Use Cases

Easily visualize, filter and store real-time multichannel neurophysiology signals. Includes built in, optimized settings for the most common biologic signal types. Use this gizmo for easy filtering and storage of common signal types: LFP, EEG, EMG, Single-Unit, EKG.

Data Stored	
Stream (optional)	Continuous filtered waveforms
Outputs	
Main	Filtered multi-channel floating point signal

Gizmo Help Slides

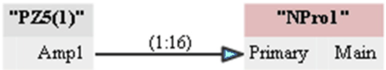
Quick Help Slide 1

Neural Stream Processor — Easily visualize, filter and store real-time multichannel neurophysiology signals. Includes built-in optimized settings for the most common biologic signal types

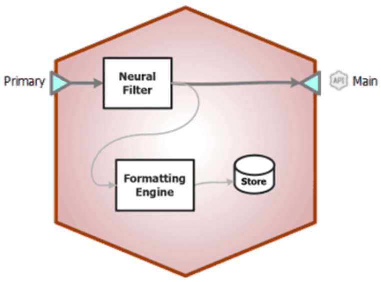
Neural data filtering of all types

EEG, LFP, EKG, ECoG, EMG, and Single-Unit filter presets

16-Channel data stream from neural amplifier directly to Neural Stream Processor for filtering and storage



Primary: Input fed from neural amplifier



Main: Outputs the filtered data stream. Data is also stored by the gizmo itself.

Quick Help Slide 2

Neural Stream Processor

User Interface, API, Data Stored

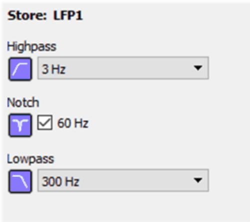
Dynamic control of filter settings during runtime.

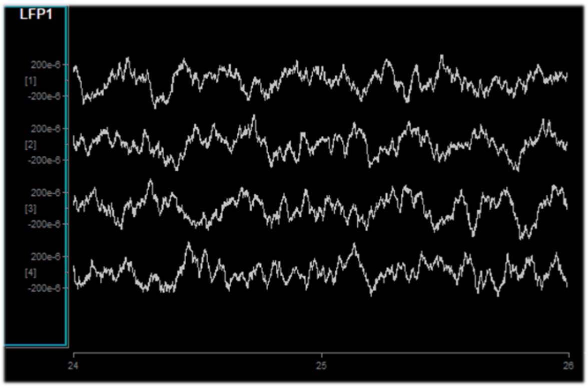
HighPassFreq — Set high pass cutoff frequency {Read/Write}

LowPassFreq — Set low pass cutoff frequency {Read/Write}

NotchFilter — Logic, enable/disable notch filter {Read/Write}

LFP1 — Filtered data stream. Name is assigned based on filtering presets but can be changed by the user. {Stream}

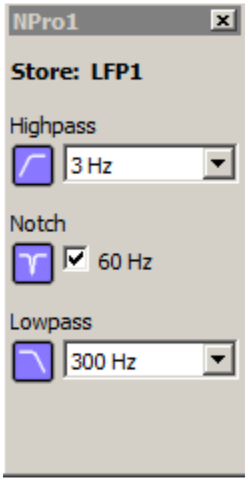




Reference

The Neural Processing gizmo takes single or multi-channel floating point signals, filters the signals and optionally formats and stores into the data tank. The filtered data can also be available as an output to other gizmos for further processing.

The Neural Stream Processor Runtime Interface



Runtime Window

Runtime Plot

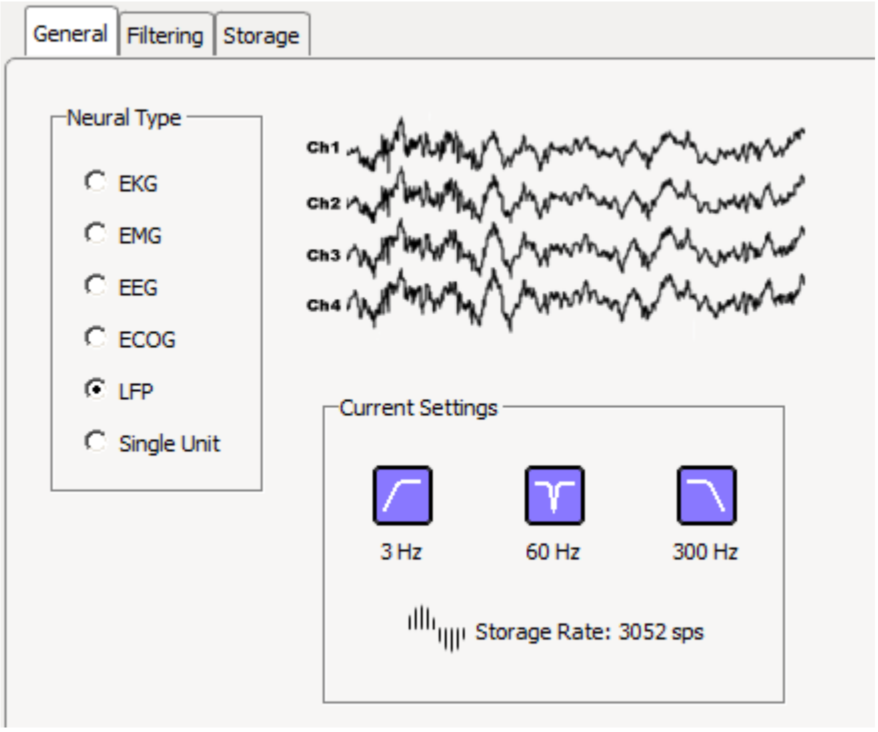
A multichannel streaming plot is included in the data plot tab when storage is enabled. See Flow Plot for more information on using and customizing the plot.

NPro1 Tab

The NPro1 tab contains controls for runtime highpass, lowpass, and notch filter adjustments, if the **Runtime Controls** option is selected at designtime.

Neural Stream Processor Configuration Options

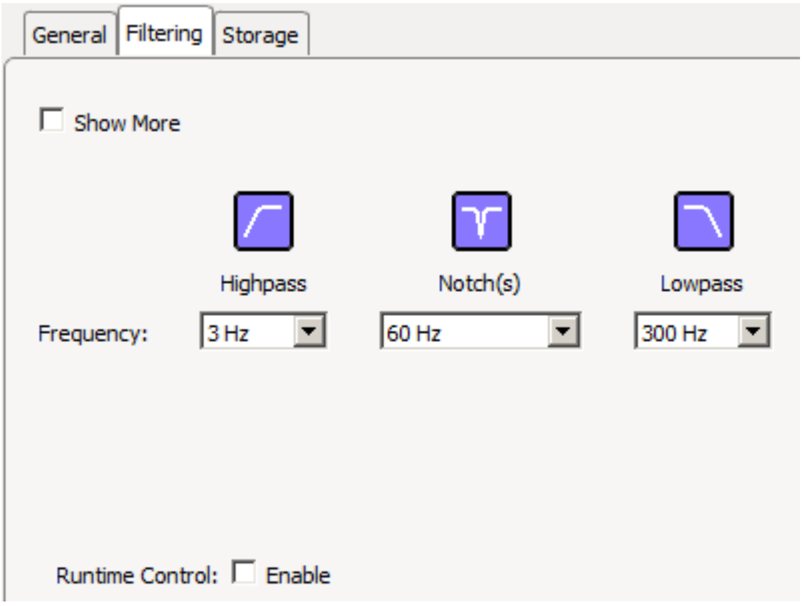
General Tab



General Options Tab

Select the signal type to automatically configure default highpass, lowpass, notch settings. A depiction of the signal type, along with the current filter and storage settings, is displayed.

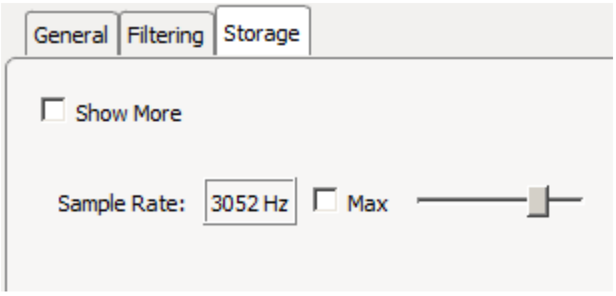
Filtering Tab



Filtering Options Tab

Select the initial highpass, lowpass, and notch filter values. To modify the highpass, lowpass, and notch settings during runtime, select the **Enable Runtime Control** check box. Click **Show More** to adjust the highpass/lowpass filter rolloff in dB/Oct.

Storage Tab



Storage Options Tab

Set the desired sampling rate of the stored data using the slider.

Click **Show More** to change the store name, data format, and scaling factor of the stored data.

Use the **Discrete Files** check box to save each channel of data as a discrete file (*.sev file) in the data tank.

Clear the **Save to Disk** check box to view data in the runtime plots without storing data to the Tank.

Oscilloscope

Common Use Cases

Has all the functionality of a hardware oscilloscope and more. View up to four channels at user-defined ranges and domains, and perform complex signal testing for creating trigger outputs. Use this gizmo to visualize signals on a more refined time scale, or to perform thresholding or hysteresis tests for complex triggering paradigms like phase-locked stimulation off LFPs.

Data Stored	
Epoc (optional)	Feature state
Epoc (optional)	When all conditions are met
Snippet (optional)	Capture the current oscilloscope waveform when all conditions are met
Outputs	
Feature	Logic signal, current feature state (pass/fail)
Trigger	Logic high when all conditions are met
Strobe	Logic high for duration of the capture window when all conditions are met
DelayedSig	Delayed version of waveform for capturing with other gizmos

Gizmo Help Slides

Quick Help Slide 1

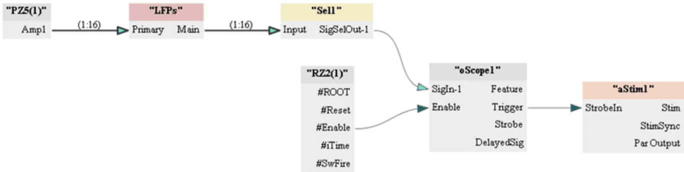
Oscilloscope — Has all the functionality of a hardware oscilloscope and more. View up to four channels at user-defined ranges and domains, and perform complex signal testing for creating trigger outputs

Phase-locked LFP stimulation

Signal monitoring at high temporal resolution

Hysteresis-tests for complex stimulation paradigms

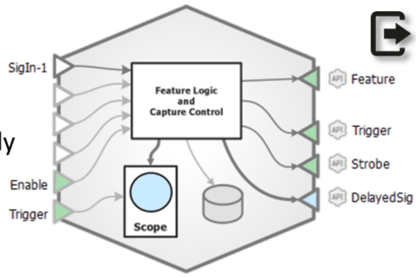
Route a single channel of LFP data to the Oscope. Trigger an auditory stimulation if the feature passes a set threshold



SigIn-:* Input signals to visualize

Enable: Master control for TTL outputs. Typically connected to #Enable

Trigger: Optional input to control oscilloscope trigger



Feature: Logic high if the feature conditions are met

Trigger: Logic high one sample when feature condition is met

Strobe: Logic high for duration of window

DelayedSig: Waveform output delayed by range + offset (no hysteresis) or range + offset + time marker 2 (with hysteresis)

Quick Help Slide 2

Oscilloscope

User Interface, API, Data Stored

Manually set oscilloscope window and feature conditions (voltage threshold and time windows for hysteresis) during runtime.

Enable — Turn the functionality of the Oscilloscope outputs on or off {Read/Write}

*Ythresh** — Change the amplitude threshold(s) for the scope feature {Read/Write}

*Xthresh** — Change the time window threshold(s) for the hysteresis feature {Read/Write}

*o1S_* — Onset/offset timestamp of when feature is true {Epoch}

*o1T_* — Onset timestamp of when all conditions are met, including hysteresis {Epoch}

Reference

The virtual oscilloscope gizmo has all the functions of an oscilloscope plus flexible trigger design tools for triggering, using more complex waveforms. Triggering is implemented on the hardware and in real-time. The gizmo's user interface provides a view into what's happening and includes controls for adjusting the signal feature threshold(s). The oscilloscope works well for simple threshold and store tasks and is an excellent tool for closed loop triggering.

The gizmo supports up to four channels of input and includes several outbound logic signals, a waveform output, and an internal data store as shown above.

The Runtime Interface

Runtime Plot

At runtime, the standard Synapse data plot is available to display any stored data. The gizmo can save epoch events when the selected feature is true and/or when the required condition passes. A snippet waveform capturing the oscilloscope plot window to disk can also be saved when triggered. These stores are selected in the oscilloscope configuration options.

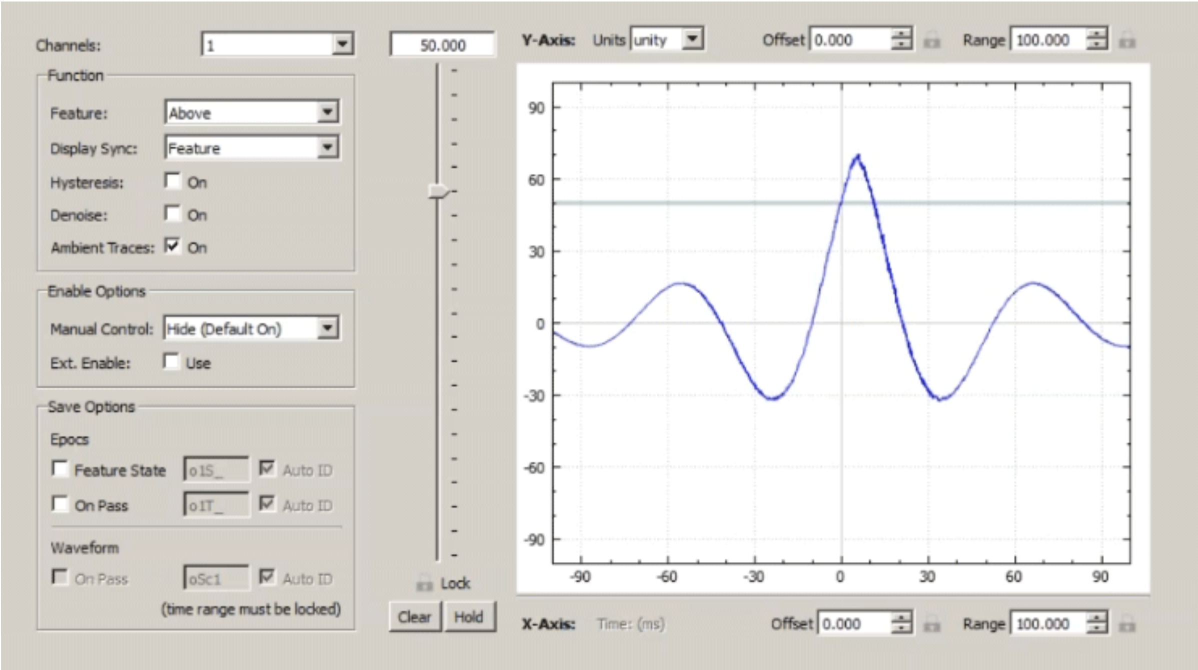
Oscilloscope Plot

The runtime oscilloscope plot must be configured in the Edit mode options before it can be used. Its use and features are nearly identical to the preview plot available in Edit mode. It allows you to modify or lock threshold, range, and offset at runtime.

> **Important**
>
> Oscilloscope is unique in that the runtime changes also modify the experiment setup. Any changes you make to the plot configuration at runtime are saved with the experiment design and vice versa. This goes against the typical Persistence model in Synapse where runtime settings are seperate from designtime settings. This helps you 'set up' the plot you want in the experiment settings using real data in a runtime mode.

Oscilloscope Configuration Options

In edit mode, the oscilloscope gizmo displays a simulated waveform alongside the gizmo options. This interface expedites setting the feature and triggering options.



Oscilloscope Options Area

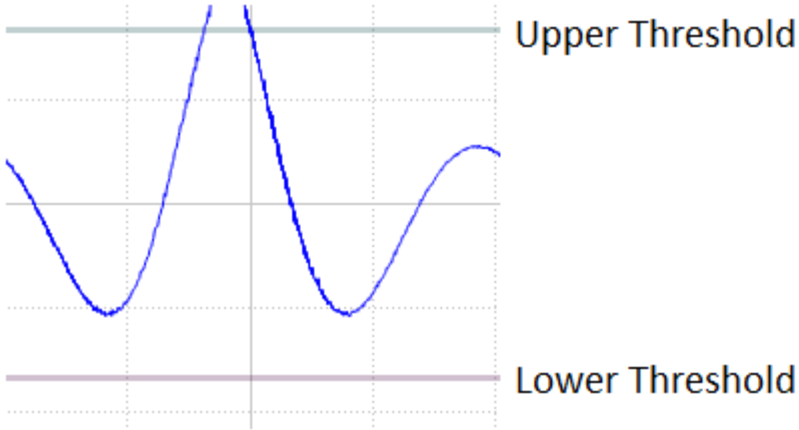
Channels

Input up to four single channels. Only one channel can be used for feature detection at a time, but when conditions are met, all channels get stored (when **Waveform** is selected).

Function

Feature

A standard threshold detection method is used to determine when a signal of interest is present. By default, the **Feature** is **Above**, and the image is synced to the feature state, just like an oscilloscope. That is, the threshold crossing is set as X=0 and the X and Y axes are set to the defined range.

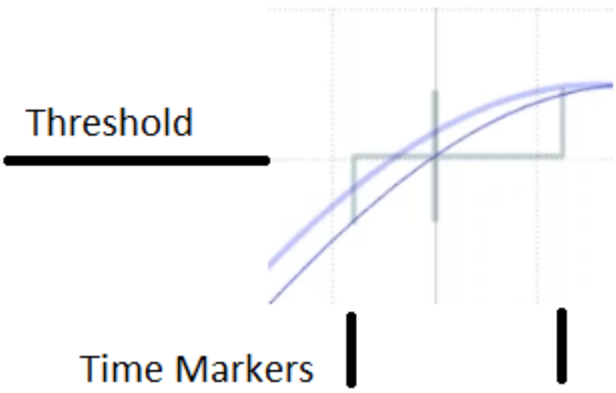
Feature Type	Description
Above	Signal is above the threshold
Below	Signal is below the threshold
Between	Signal is between an upper and lower threshold (when selected a second threshold marker is added to the plot) 
Outside	Signal is outside an upper and lower threshold (when selected a second threshold marker is added to the plot)
Rising	Signal is increasing in value
Falling	Signal is decreasing in value

Display Sync

Determines how to align the waveform at x = 0.

Option	Description
Feature	When selected feature is true
Ext Input	Trigger input from another gizmo or digital input (defined in the block diagram)
None	Not set, shows the free running traces (ambient traces)

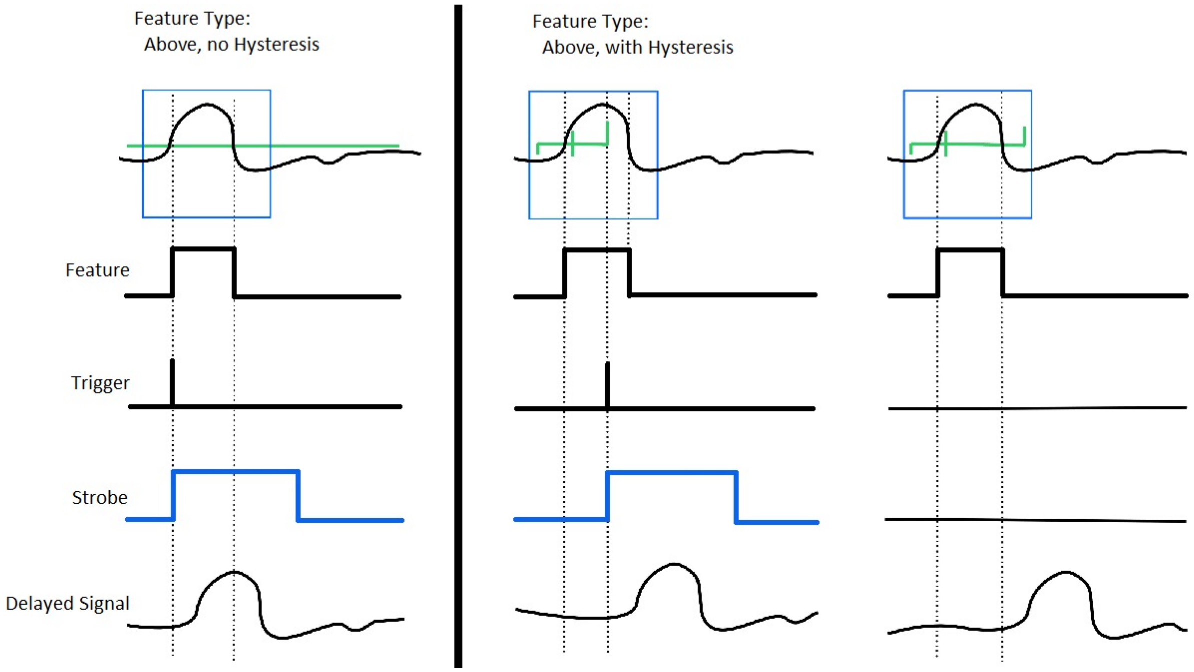
Hysteresis



To add temporal characteristics to feature detection, enable Hysteresis. The hysteresis marker replaces the default threshold marker. Two time markers are added on the x-axis and are controlled by sliders.

In the illustration at right, the feature condition must be false for the time period between **Time**

Marker 1 and the threshold crossing, and must be true between the threshold crossing and **Time Marker 2** to pass all conditions. If it passes, the Trigger output pulses high for one sample. If it does not pass, the candidate waveform is displayed as a thick light blue line.



As shown in the diagram above, when Hysteresis is used, the signal must meet all conditions or no trigger is fired.

The Delayed Signal output is shown at the bottom. When there is no Hysteresis, this signal is delayed by range + offset, so you are always looking at the waveform window when the Strobe is high. When you do use a hysteresis, the delay is the range + offset + time marker 2.

Denoise adds a fixed three sample debounce to prevent spurious feature detections in a noisy signal. The feature must be true for 3 samples to register as a valid event.

Ambient Traces option show all traces even the ones that don't meet feature conditions. They are shown as thinner lines. If you don't have any signals meeting the feature conditions, viewing ambient traces can show where you need to set the threshold.

Enable Options determine the state of the manual controls at runtime. When the controls are on an **Enable** button is added to the runtime window. When present, it must be selected to enable the Trigger and Strobe outputs.

When the option is set to hide (default on), the Trigger and Strobe outputs are enabled (On), but the button is hidden. For more on the Strobe and Delayed Signal outputs see "Storing Outputs" below.

Save Options

Epocs

- **Feature State** stores the Feature state on/off timestamps. Feature state remains true/high (1) as long as the Feature condition is met.
- **On Pass** stores the Trigger timestamp. Trigger fires once when all conditions are met, including any hysteresis.
- **Waveform** On Pass store the plot snippet when feature conditions are met.

Storing Outputs

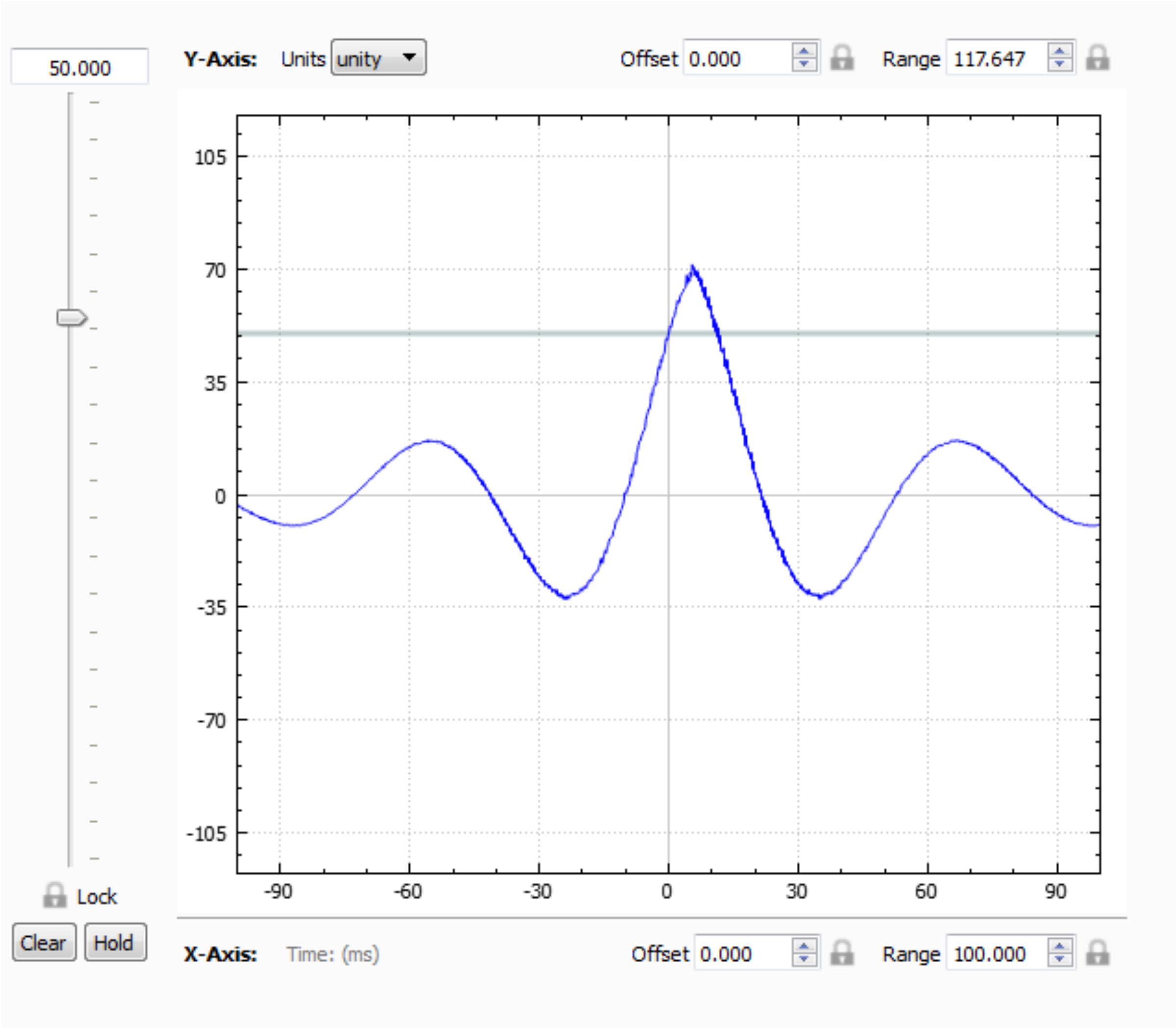
You can also pass the oscilloscope outputs (seen in the block diagram) to storage gizmos.

If you wanted to save the waveform on pass, but didn't want to use a fixed x-axis window (range), you can send the Strobe and DelayedSig output to a Strobe store gizmo.

- **Strobe** starts when Trigger fires (all conditions are met) and remains high for the duration of the X-axis window (range).
- **DelayedSig** is a continuous waveform delayed by the X-axis range and offset, such that you can always store what you are seeing in the plot.

Preview Plot Options

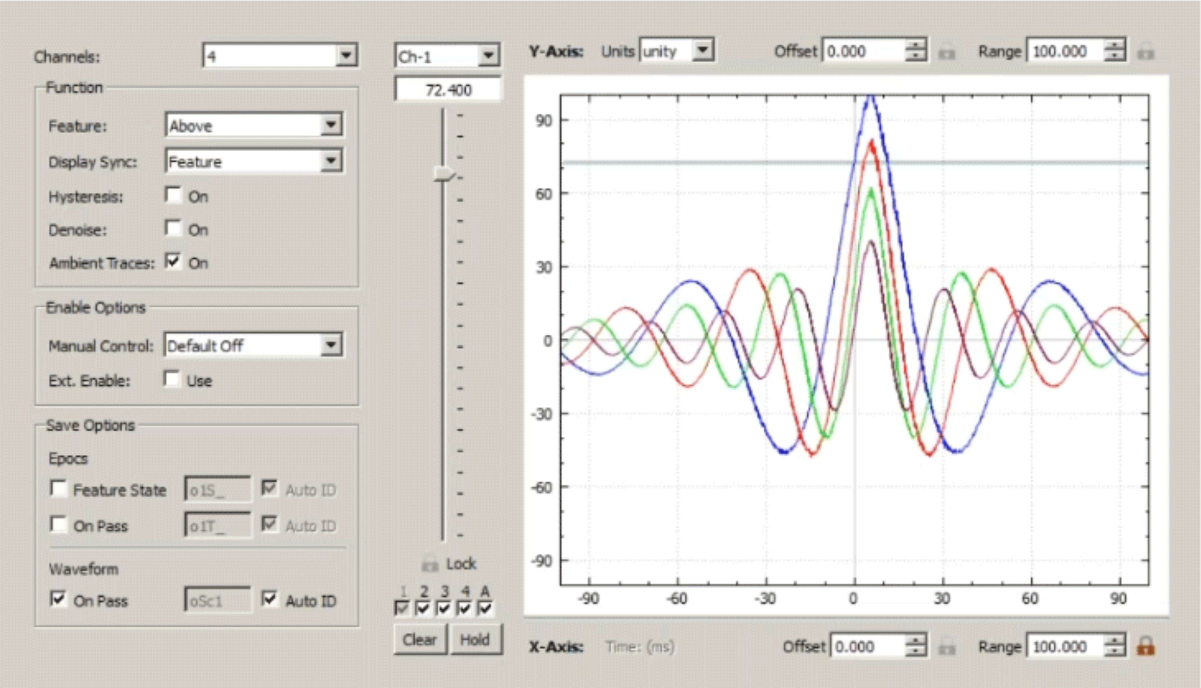
The plot area includes intuitive controls and provides immediate visualization of your changes. Drag the threshold marker along the slider to set a threshold.



Single Channel Preview Plot

Using Multiple Channels

When you use multiple channels, you can select which channel is used for detection by selecting from the drop-down menu located above the threshold slider. You can also choose which channels to display in the plot, using the channel number check boxes below the threshold slider.



Multi-channel Preview Plot

Check boxes for each channel are shown below the threshold slider. The grayed check box is the currently selected channel. Select additional numbered check boxes to show those channels in the plot.

Plot Controls

Description	Shortcut
Move Offsets	click-and-drag
Y-axis Zoom	mouse-wheel
X-axis Zoom	Ctrl + mouse-wheel

Parameter Manifold

Common Use Cases

Control multiple stimulation gizmo parameters simultaneously. Use this gizmo when needing to share parameters between multiple stimulation gizmos, such as duration or pulse count. Often used in conjunction with the Parameter Sequencer gizmo.

Data Stored	
Epoc (optional)	Parameter values when triggered
Outputs	
Main	Logic trigger
ParOut-1..4	Parameter streams to connect to stimulation gizmos
SCout-1..4 (optional)	Single channel, floating point parameter values

Gizmo Help Slides

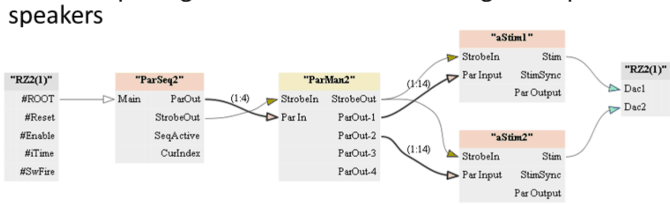
Quick Help Slide 1

Parameter Manifold — Control multiple stimulation gizmo parameters simultaneously

Coordinate timing and parameter values of multiple stimulation gizmos – DPOAEs, contralateral auditory stimulation, complex electrical stimulation

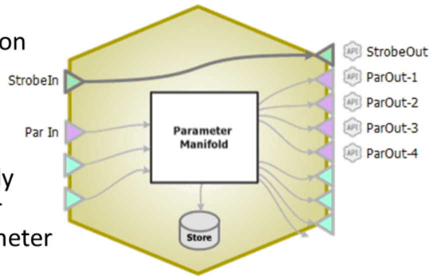
Control User Gizmos values utilizing the Parameter Sequencer

The Manifold routes shared (Level) and unique (Frequency) parameters to separate audio stim gizmos, which output signals on Dac1 and Dac2 to go to separate speakers



StrobeIn: Input to control presentation timing. Manual strobing is also available

Par In: Dynamically control parameter values from Parameter Sequencer



StrobeOut: Strobe signal to activate stimulation gizmos

ParOut:* Parameter values for each child gizmo for the given presentation

Quick Help Slide 2

Parameter Manifold

Gizmo Configuration, Data Stored

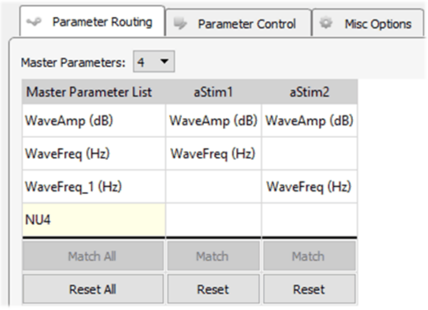
Select which parameter outputs go to each child gizmo in the Parameter Routing Table

Master Parameters — Parameters for child gizmos appear here.

Determine which ones are going to be shared by putting them in the same row.

Double-click the NU* parameters to create your own parameter. This is most useful for creating parameters to pass to for User Gizmos

Epoc — Optionally store the parameters on stim onset {Epoch}



Parameter setup used in example experiment on previous slide

Note* Child gizmos must be attached with parameters set to *Mode: Param In* for the Manifold to recognize them

Quick Help Slide 3

Parameter Manifold

Parameter Table

This gizmo has a Parameter Table to dynamically control values its parameters from other gizmos and during runtime

Parameter Tables provide the user dynamic control over parameter values. Each one of the Modes will change how the user can control and interact with the parameter.

Param In – Dynamically control parameter values from Parameter Sequencer or Parameter Manifold

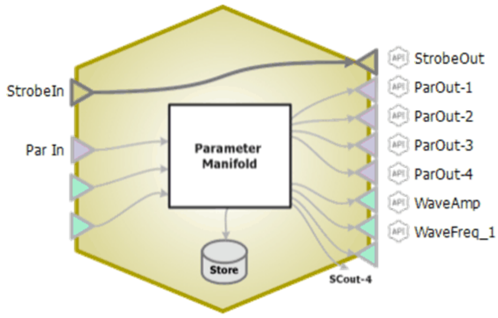
Constant – The value is immutable at runtime

Scalar In – Pass a scalar value from another gizmo for dynamic real-time control

Widget – Creates a widget (slider) on the user interface that is controlled manually by the user or through the API.

Scalar Out (Scout) – Makes this value available as a gizmo output to connect to another gizmo input in real-time

Epoc – Save the parameter values used for each stimulus presentation



Changing the Mode of parameters will change the input and output options on the gizmo.

All parameters can be read through API. Only parameters in Widget mode can be written to.

All parameters are connected to the “Par Output” gizmo output link

Parameter Routing | Parameter Control | Misc Options

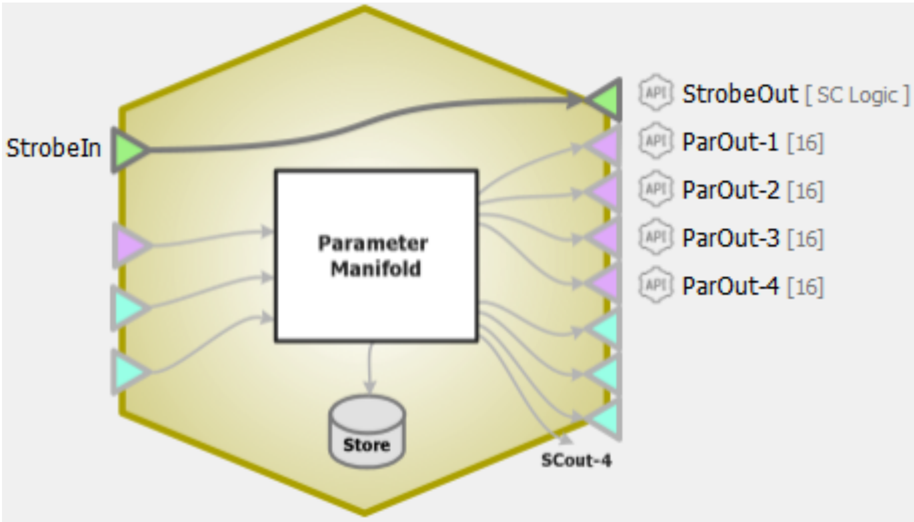
Master Parameter Set: Show All

	Name	Mode	Value	Jit(%)	Min	Max	Epoc	ID	Auto ID	SCout-1
1	WaveAmp (dB)	Param In	0.0	0.0	-120.0	120.0	on Stim	Wamp	☑	◉
2	WaveFreq (Hz)	Param In	0.1	0.0	0.1	100000.(	None	Wfrq	☑	○
3	WaveFreq_1 (Hz)	Param In	0.1	0.0	0.1	100000.(	None	Wfrq	☑	○

Reference

Use the Parameter Manifold if you have multiple stimulation gizmos that require parameter inputs that you want controlled from the same Parameter Sequencer gizmo. Each stimulation gizmo brings its own parameter list into the manifold. Parameters used in multiple gizmos can retain individual values or use a common/shared value. For example, two stimulation gizmos might use a common PulsePeriod, but different WaveAmps.

You can also use the Parameter Manifold to create parameters for a User Gizmo.

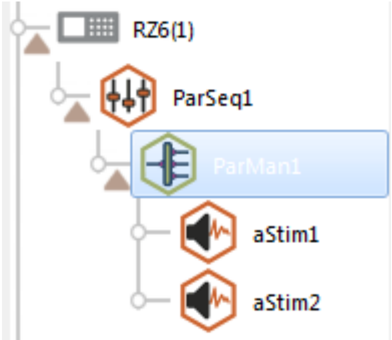


Parameter Manifold Block Diagram

Adding a Parameter Manifold to Your Experiment

This gizmo links between the stimulation gizmos and the Parameter Sequencer gizmo in the signal/processing path. To establish the links, the gizmos rely on input/outputs that must be configured in other gizmos in this path. Because of this, you will need to follow the ordered steps below:

1. Add your stimulation gizmos to the Processing Tree. You can temporarily add them to the stimulation device. They will be moved later.
2. Set the parameters, choosing **ParamIn** for any parameters you want to automate/control using the manifold.
3. Add the **Parameter Manifold** to the Processing Tree. You can temporarily add the manifold to the stimulation device. It will be moved later.
4. Connect the stimulation gizmos to the Parameter Manifold. They will inform the manifold of the parameters they need from it.
5. Configure the Parameter Manifold (see below).
6. Add the Parameter Sequencer to the stimulation device, such as an RZ6.
7. Connect the Parameter Manifold to the Parameter Sequencer. It will inform the sequencer of the parameters it needs.
8. Configure the Parameter Sequencer.



Parameter Manifold in the Processing Tree

Important

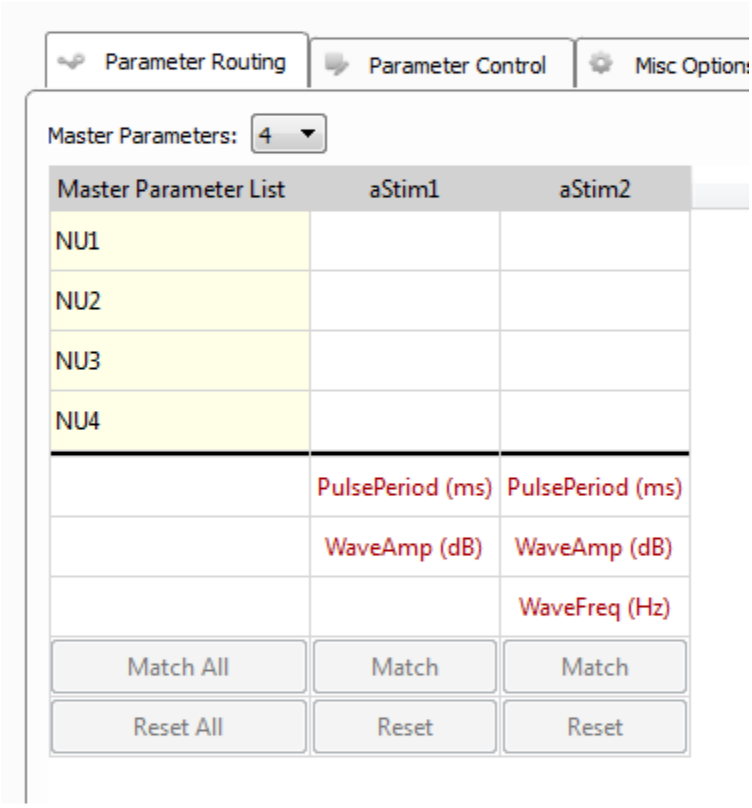
Each gizmo linked to the manifold must be attached to a unique output (ParOut-1..4). The numbered outputs match the indexed columns in the routing/matching table (see below) and this information is used to populate the master parameter table.

Parameter Manifold Configuration Options

Parameter Routing Tab

The manifold pulls together the parameters used by each gizmo and generates a master parameter list. The master list and parameters are auto-filled in a table on the Parameter Routing tab. Before the parameters from each gizmo are matched to a master parameter, they are organized in columns, shown in red, and filled below the main table rows.

In this example, two Audio Stimulation gizmos are attached to the Parameter Manifold. The columns contain the gizmo names and the rows contain the parameter that the manifold is controlling. aStim1 is a noise stimulus and aStim2 is a tone stimulus. They have some, but not all, parameters in common. Initially, the parameters for each gizmo are unassigned and appear in red text at the end of the column.

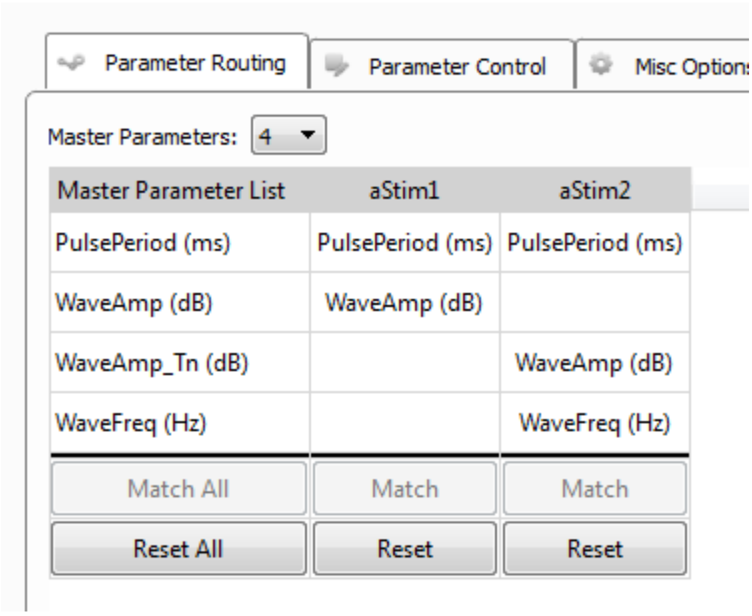


Master Parameters Matching Table with Unassigned Parameters

You can double-click, drag, or use the Match and Reset buttons to move the unassigned parameters into master parameter rows. As you do this, Synapse auto fills the parameter names in the master column.

If a parameter is present in more than one gizmo, but will NOT share a common value, you might need extra rows. Use the Master Parameter drop down list to increase the number of rows (if needed). When you have enough rows, drag one of the duplicate parameters into an unused row. Double-click the first cell in the master row to give the master parameter a different name.

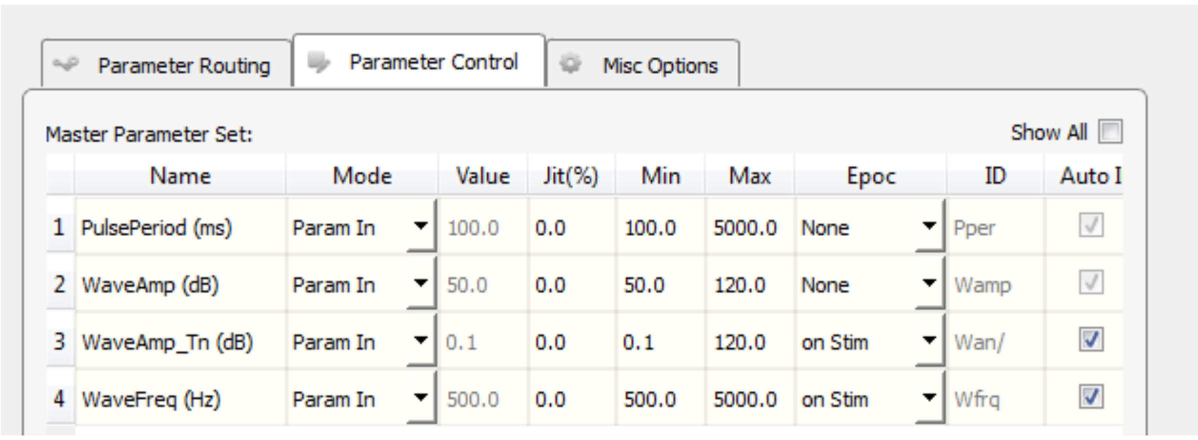
In the illustration below, the tone and noise stimuli share a common pulse period (PulsePeriod, shown in the first row). Frequency (WaveFreq) is used in the tone stimulus (aStim2) but not for noise (aStim1). Amplitude (WaveAmp) is used in both, but the value will not be shared. A new parameter, called WaveAmp_Tn, has been created to differentiate the tone amplitude from the noise amplitude.



Master Parameters Matching Table with Shared and Individual Parameters

Parameter Control Tab

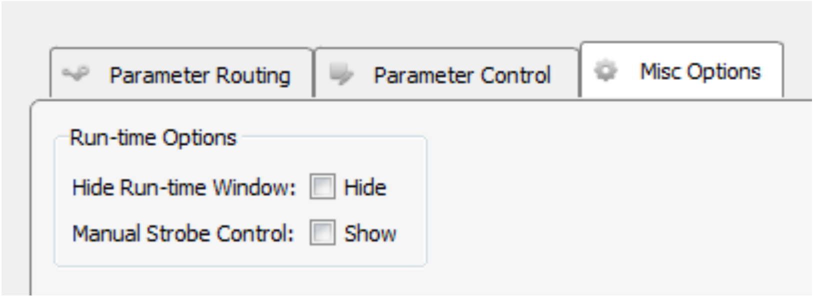
The Master Parameter Set table on the Parameter Control Tab functions like any other parameter table (see Using Parameters). We will control the parameters from a Parameter Sequencer gizmo, so the Mode for the parameters must be set to **ParamIn**.



Parameter Control Tab

Misc Options Tab

Use Run-time Options to show and hide run-time features.



Misc Options Tab

Parameter Sequencer

Common Use Cases

Control stimulus parameters with complex timing and presentation sequences (rolling, repeated, random, manual). High-level parameter control and stimulus presentation.

Data Stored	
Epoc (optional)	Parameter values when triggered
Outputs	
ParOut	Parameter streams to connect to stimulation gizmos
StrobeOut	Logic signal to trigger stimulus
SeqActive	Logic signal, true when sequencer is running
CurIndex	Integer, currently presented parameter row index

Gizmo Help Slides

Quick Help Slide 1

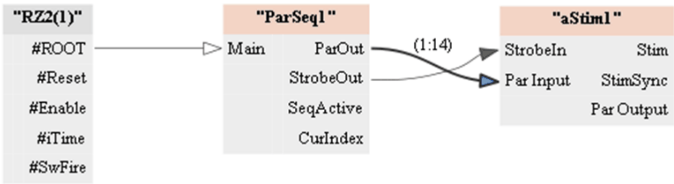
Parameter Sequencer — Control stimulus parameters with complex timing and presentation sequences (rolling, repeated, random, manual)

Oddball paradigm (P300) experiments

Drive stimulation gizmos with complex patterns

Behavioral trials that include stimulation presentations

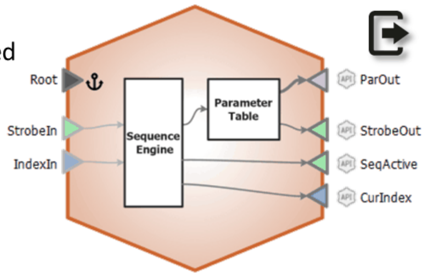
ParSeq1 (default tied to #ROOT) sending strobe output and parameter values to an Audio Stimulation gizmo



Root: Anchored to the processor. Does not need inputs

StrobeIn: Optional input to control presentation timing

IndexIn: Optional input to control which sequence row to present



ParOut: Parameter values for the given presentation

StrobeOut: Strobe signal to activate stimulation gizmo

SeqActive: Logic high during presentation sequence

CurIndex: Integer value for the current sequence row

Quick Help Slide 2

Parameter Sequencer

User Interface, API, Data Stored

Run, pause, or manually strobe presentation sequence. Change timing and presentation jitter.

Automated — Set Sequencer to run in Automated or Manual mode {Read/Write}

Reset — Initiates an automated sequence {Read/Write}

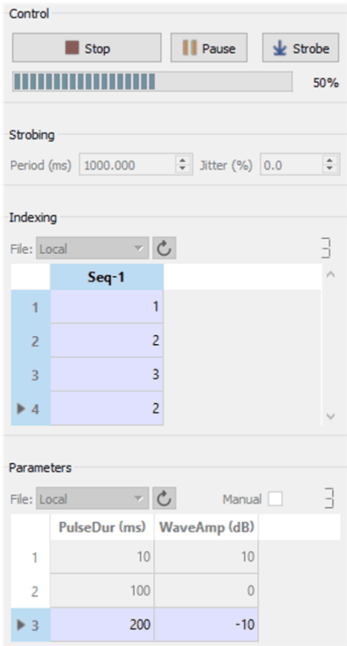
Progress — Number of presentations that has occurred since sequence start {Read}

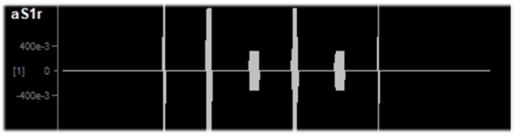
ManualIndex — In Manual mode, integer that sets the next presentation row {Read/Write}

Strobe — In Manual mode, present the currently selected row {Write}

*Seq-** — All sequences and parameters can be dynamically written arrays {Read/Write}

None





Reference

The Parameter Sequencer gizmo is an interface for controlling stimulus parameters and presentation sequences. It is a highly flexible gizmo with many options for timing, triggering, and control to cover a wide variety of stimulus presentation needs. It is typically used with any of the stimulation gizmos. By selecting the **Param In** mode in the stimulation gizmo's parameter table you tie that parameter to the parent Parameter Sequencer.

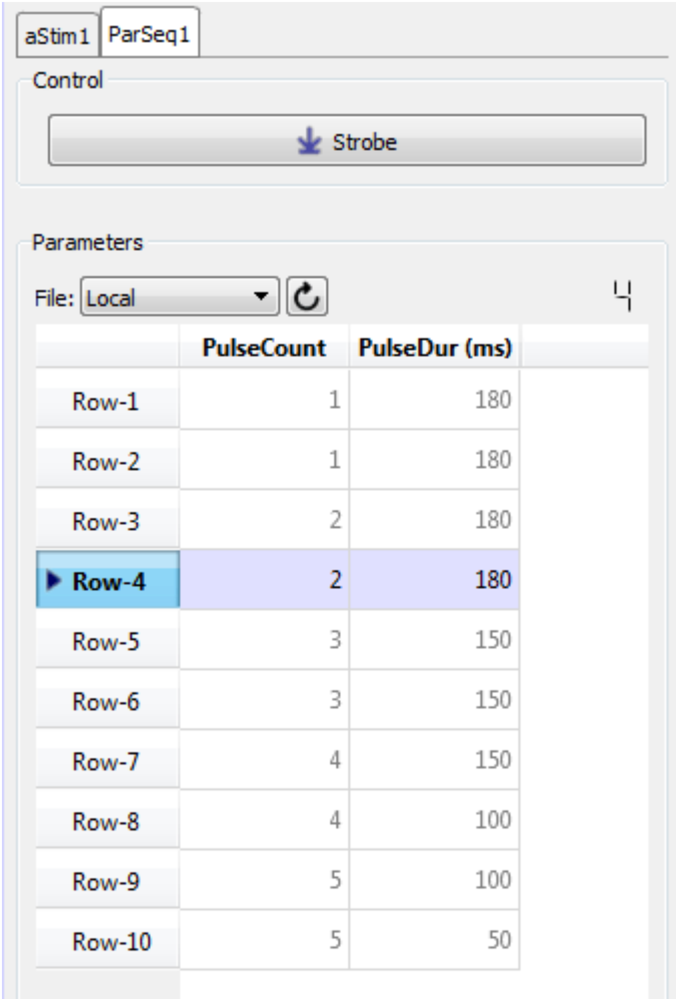
Example

See Using Parameters for more general information on parameters and experiment walkthrough examples.

The Parameter Sequencer requires no inputs by default. Optional inputs for presentation timing and row selection are available depending on gizmo selections. Several different types of outputs are available for monitoring, storage, or to trigger other gizmos or devices.

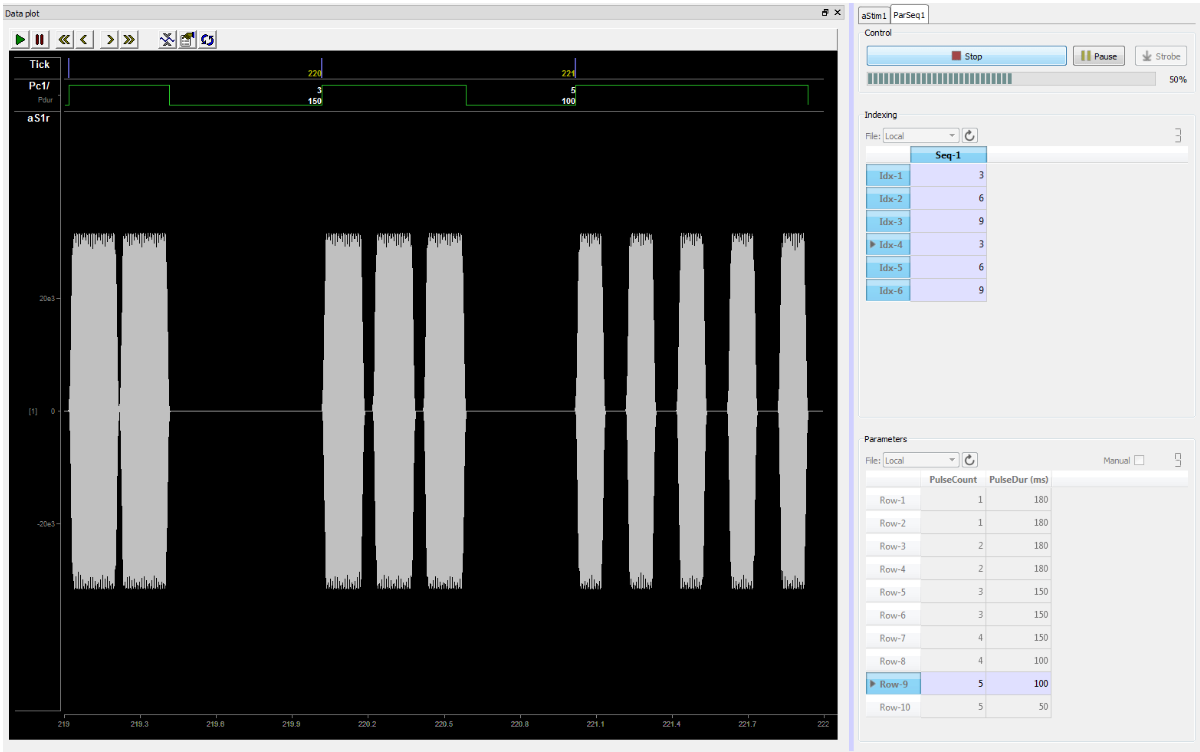
Parameter Sequencer Runtime Interface

At its most basic, the sequencer runtime interface is a table of stimulus parameter combinations that can be selected for use at runtime.



Parameter Sequencer Tab, Split Window

In the example above, combinations of values for Pulse Count and Pulse Duration are saved to the database and manual control was selected. Here you can trigger a presentation of the stimulus by clicking the **Strobe** button. The parameters in the highlighted row will be used to generate the stimulus. You can change rows by double-clicking a value cell in the desired row. Once the new row is selected, click **Strobe** again.



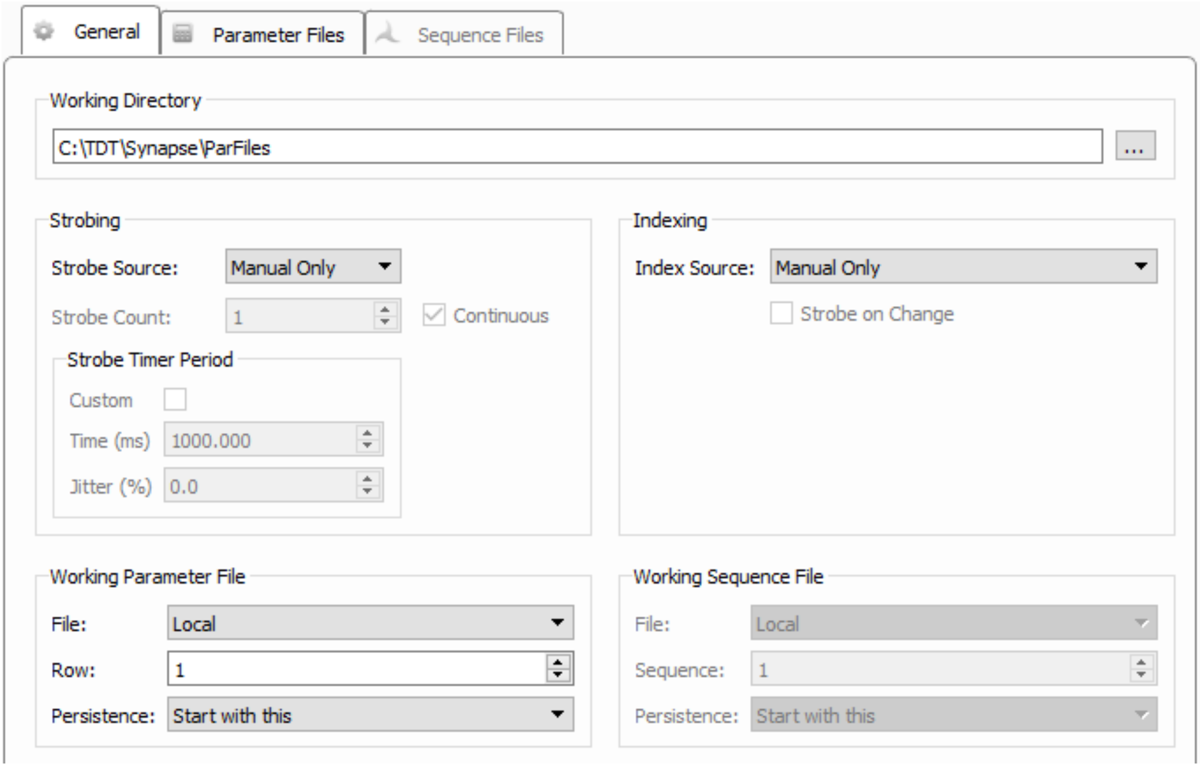
Parameter Sequencer Used to Manually Control Progression of Audio Stimulation Through a Sequence of Tone Bursts

The stimuli can also be presented with the parameter combinations in the order listed in a sequence file. The example above shows the first three indexes (1 - 3), which correspond to rows 3, 6, then 9 in the parameters table. There is a visible difference as the pulse duration and count changes with each presentation. In this example, the interface is designed so you click play, then it runs through the sequence in response to a strobed input line. With many configurable options and the ability to combine the Parameter Sequencer with any of the stimulus gizmos and the Parameter Manifold gizmo, the possibilities will cover most any situation.

Parameter Sequencer Configuration Options

General Tab

The General tab pulls together information about where related files will be stored and how the sequencer will be timed/triggered.



General Tab

Working Directory

By default, the parameter and sequencer tables are stored with the experiment. But you can also save them locally to disk as CSV files so you can use the same parameters across multiple experiments, share them, or modify them outside of Synapse. Sequencer related files are stored in the folder you designate as the Working Directory; by default, C:\TDT\Synapse\ParFiles. If the address entered does not exist, the field will be highlighted in red. You can use the **Browse[...]** button to navigate to the parent folder and create the desired folder.

Strobing

Strobing determines how the sequencer advances. The strobe can be based on one of three possible sources:

1. If **Manual Only** is selected, you must generate the strobe manually using the **Strobe** button on the runtime interface.
2. **Strobe In** controls the sequencer with a logic signal connected to the gizmo's StrobeIn input. **Strobe Count** and **Continuous** options provide duration control.
3. **Timer** defines an internal timer to control the strobe, using **Time(s)** and **Jitter(%)**. Use jitter to introduce a random variability to the period for your timer.

Check the **Custom** checkbox when you are using the Sequencer (**Index Source** is set to **Sequence File**) to enable an additional Time column in the sequence file that controls the specific time, in milliseconds, between each presentation for fully custom timing.

Working Parameter File

In this area you must select the parameter file you will be using. **Local** is the default selection which is the parameters that are saved with the experiment. If you want to start the experiment with a different file loaded, select it here.

Parameter files are created on the Parameter Files tab.

Using Row and Persistence you can determine where in the parameter file presentation will begin and if you want to start there or lock persistence to that row.

Indexing

The Indexing Source determines how you advance through the sequence of parameters. You can advance the index manually, using a gizmo input, or from a sequence file. The **Manual Only** and **Gizmo Input** options behave similar to the same options for the strobe source, described above. With a gizmo input, you have the option of sending a strobe out to indicate the change in the index. **Strobe Out** can be stored or used to trigger other gizmos. You also get the option to automatically start the sequence presentation when the recording begins (**Start Seq at Run-time**).

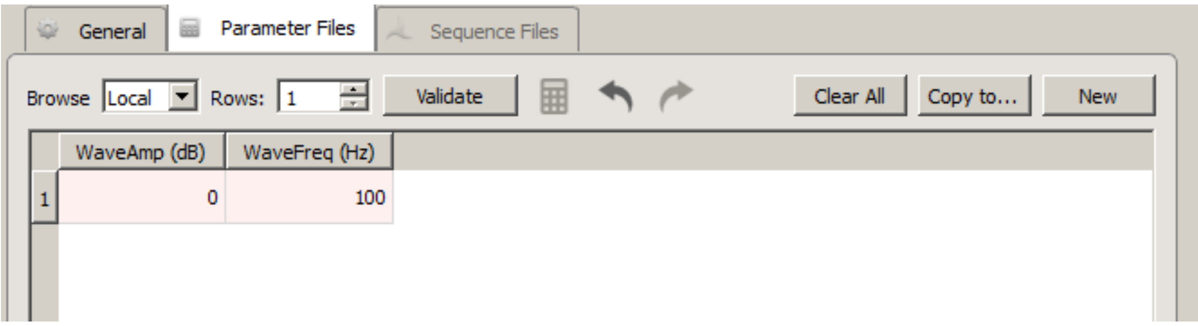
Sequence File

Choose how the parameter index is controlled.

Sequence files are created on the Sequence Files tab.

You can optionally start the sequence presentation automatically when the recording begins (**Start Seq at Run-time**), and optionally stop the recording at the end of the sequence (**Idle When Done**).

Parameter Files Tab



Parameter Files Tab

This tab is a visual interface for selecting or creating a list of all combinations of parameters that you want to use with the stimulation gizmo at runtime. When the gizmo is linked to a stimulation gizmo in the Processing Tree, the parameters are automatically added to the table as columns.

Values entered in the table are checked against the parameter's minimum and maximum, as defined in the stimulus gizmo parameter table. This check is made automatically when the gizmo options are committed and can also be made by clicking the **Validate** button. The values you enter on the Parameter Files tab serve as the **Local** file. These values are saved within the gizmo as part of the current experiment and are not saved in a separate file. Changes made to the current file overwrite any previous values.

Saving a Copy (Copy To, New)

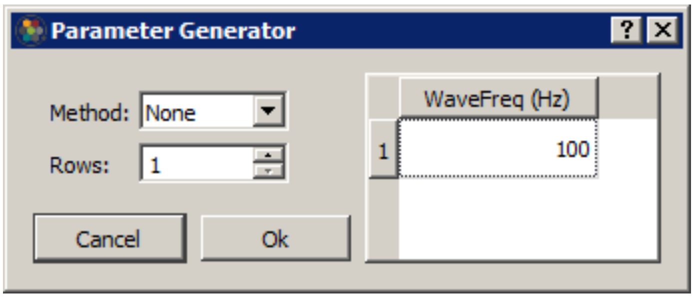
You can save a copy of the current parameter table as a CSV file (*.par.csv) in the working directory defined on the General tab. This ensures a permanent copy of the parameter set and allows you to have more than one file, reuse, and share files across experiments.

Use the **Copy To** button to save parameter sets that have already been filled into the table. Once the CSV file is created any changes in the table are saved to the CSV file using the gizmo's **Commit** button.

Use the **New** button to create a new blank CSV file. The empty file is created immediately, but it will not be filled until parameters have been added and committed.

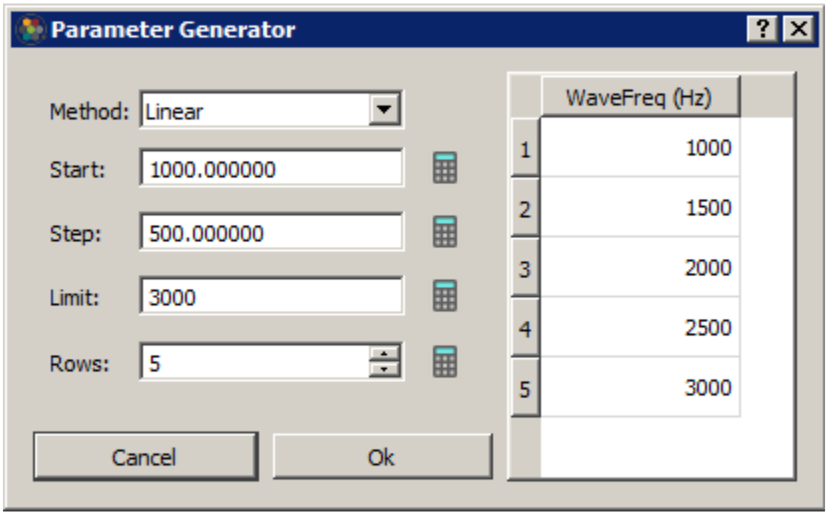
Parameter Generation

To make this whole process easier, Synapse can automatically build out a parameter list using common mathematical operations. Select a column in the table and click on the calculator button to open the Parameter Generator dialog.



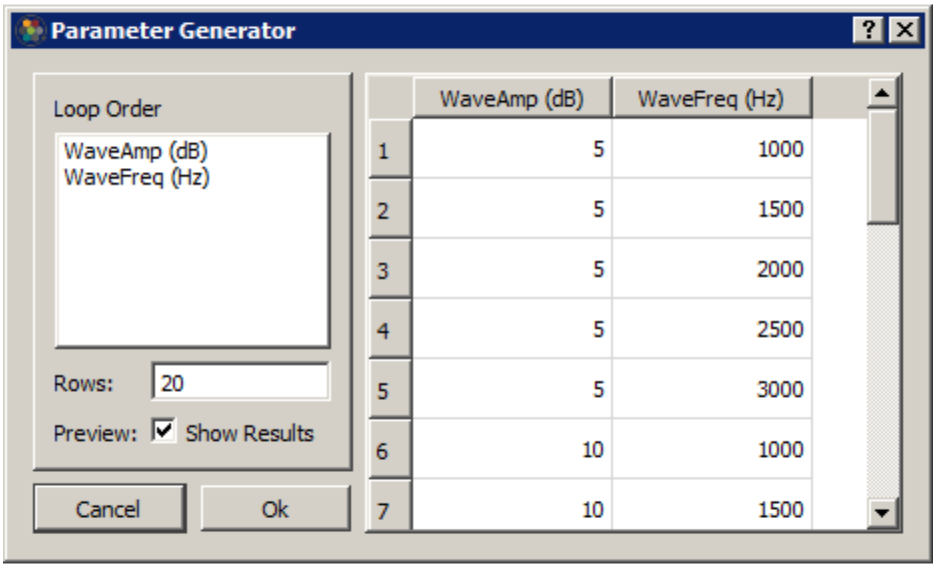
Generation Methods include **Constant**, **Linear**, **Log2**, **LogN**, **Log10**, **Random**, and **Gaussian****.

For methods that use a mathematical equation (**Linear**, **Log2**, **LogN**, **Log10**), set any of the three parameters and click the calculator button next to the fourth parameter to automatically generate it.



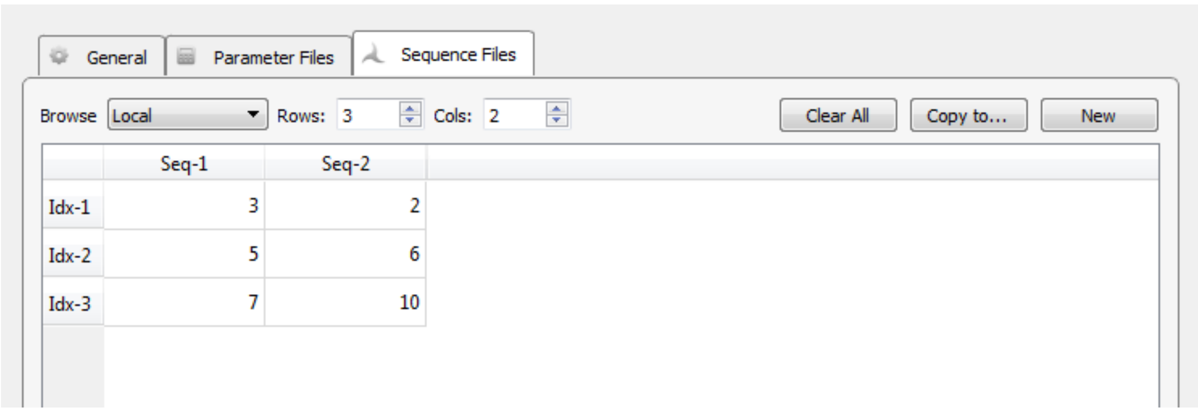
When you click OK those parameters are automatically added to the table.

If you want to generate all possible combinations of parameters in your table, select multiple parameter columns and click the calculator button to generate the combinations.



Sequence Files

This tab is enabled when the **Index File** option is selected on the General tab. The tab functionality is similar to the Parameter Files tab, described above. You can create a new sequence file or browse to an existing one and you will find the same **Copy To** and **New** button options for working with multiple sequence files.



Sequence File Tab

In the sequence table, a column represents a sequence of stimulus presentations, with each index, or row, pointing to the desired set of parameters, or row of the parameters table, for that presentation. You can rename the column headers for your experiment. The illustration above shows a value of 3 for the first index of the first sequence. That means that the parameter values in the third row of the parameter file table will be used. When signal presentation advances to the second index in the first sequence, parameter values will be pulled from row 5 of the parameter file table.

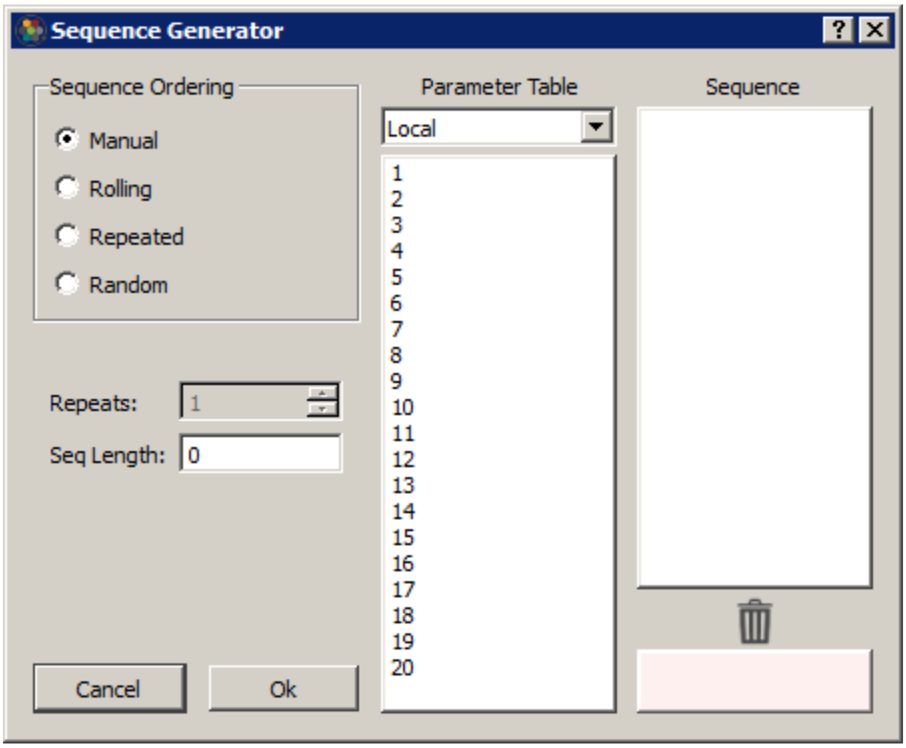
> **Tip**
>
> You can rename the parameter rows and reference them by name in the sequence column, for example, double-click on a parameter row index number and rename it to "ToneA", and then in the sequence file enter "ToneA" in the column.

When you have more than one sequence, you can choose which sequence to begin with or to lock to on the General Tab.

If the **Custom** checkbox is selected on the General Tab, an additional **Time** column for each sequence will be available to set the specific timing of each row of stimuli.

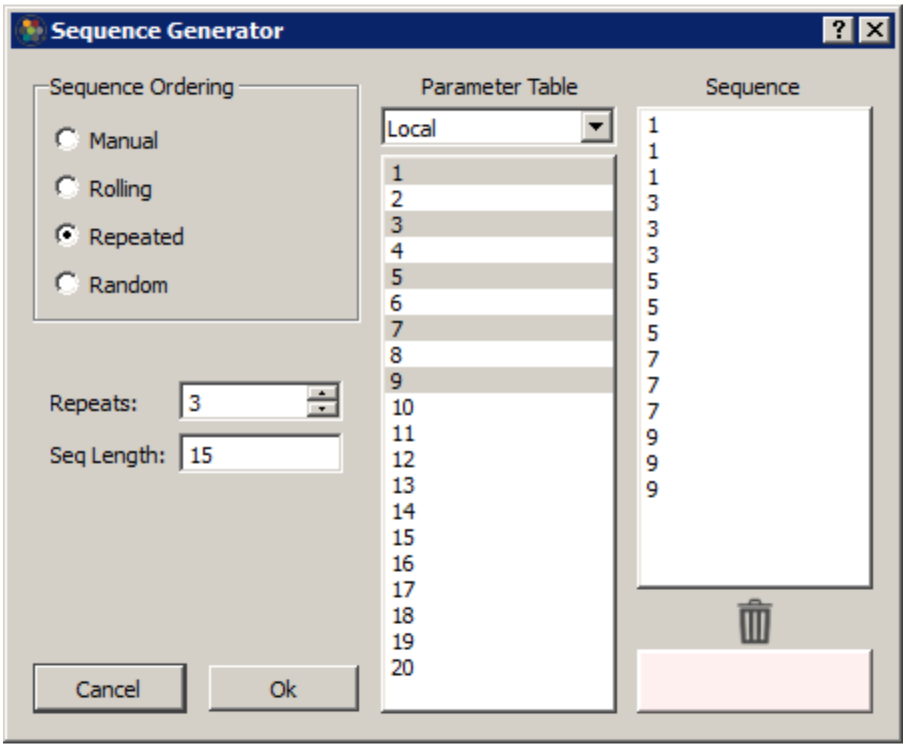
Sequence Generation

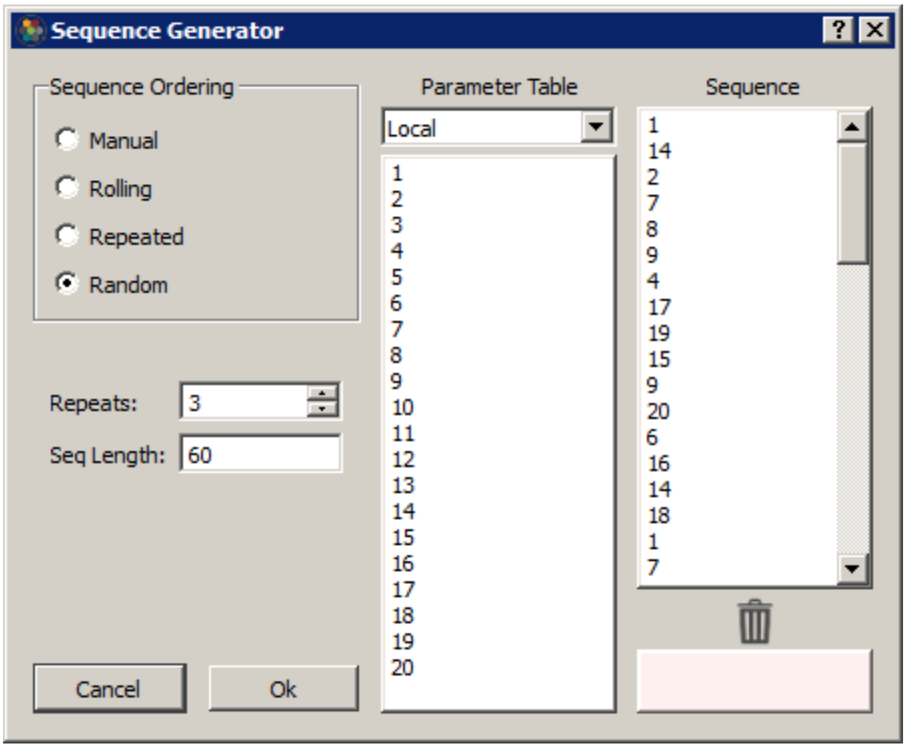
Synapse can automatically build out a sequence list using common ordering operations. Select a column in the table and click on the calculator button to open the Sequence Generator dialog.



In Manual mode, you can drag/drop rows into the Sequence column to order them. Drag/drop from the Sequence list to the trash list at the bottom to remove from the sequence.

In the other modes, select the number of Repeats and the Sequence list will automatically generate using the selected rows in the Parameter Table list (it will use all rows if no rows are selected). Some examples:


Sequence Generator
Sequence Ordering
Manual
Rolling
Repeated
Random
Repeats: 3
Seq Length: 15
Parameter Table
Local
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Sequence
1 1 1 3 3 3 5 5 5 7 7 7 9 9 9
Cancel
Ok


Sequence Generator
Sequence Ordering
Manual
Rolling
Repeated
Random
Repeats: 3
Seq Length: 60
Parameter Table
Local
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Sequence
1 14 2 7 8 9 4 17 19 15 9 20 6 16 14 18 1 7
Cancel
Ok

PCA Spike Sorting

Common Use Cases

Real-time filtering, spike detection, and principal component-based spike sorting with selectable algorithms. This is the most common method for online spike sorting. Cluster units in PCA space and identify spikes automatically or manually cut.

Data Stored	
Snippets (optional)	Timestamped spike waveforms
Stream (optional)	Plot decimated waveforms
Outputs	
Main	Filtered, multi-channel floating point signal
Sort Codes	Multi-channel integer signal containing compressed sort codes

Gizmo Help Slides

Quick Help Slide 1

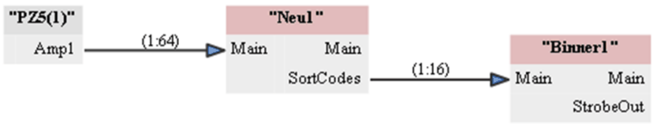
PCA Spike Sorting — Apply real-time filtering, spike detection, and principal component-based spike sorting on neural signals from your amplifier

Neural recording with action potentials

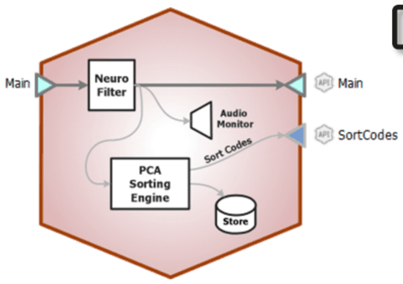
Acute neurophysiology

Chronic behavior experiments

64-Channel PCA spike sorting from raw amplifier data stream, sort codes to Sort Binner



Main: Input fed from neural amplifier



Main: Filtered signal output. Typically unused because data stream can be saved directly in gizmo

SortCodes: Multi-channel integer values containing sort code information. Typically goes to Selector or Sort Binner for further online processing

Quick Help Slide 2

PCA Spike Sorting

User Interface, API, Data Stored

Dynamic control of filters, thresholding, and clustering allows for advanced clustering of thresholded snippets.

API *FreqHP/FreqLP* — Corner frequency control of filters {Read/Write}

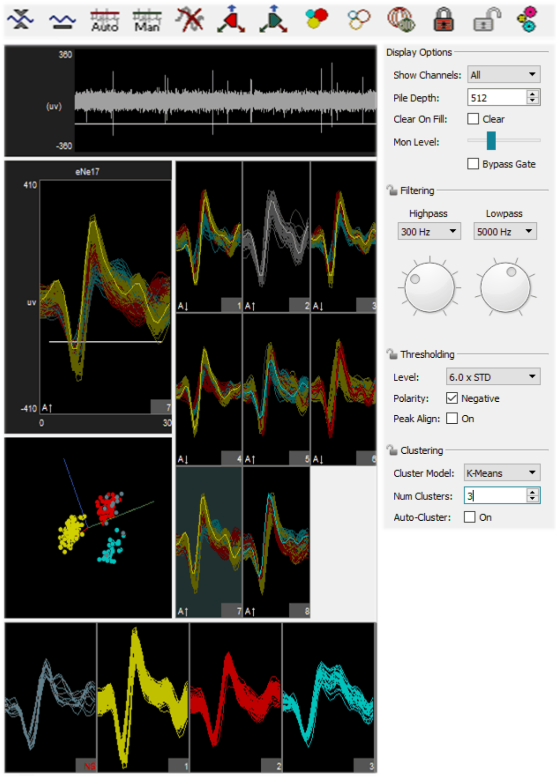
ThreshLock — Lock the threshold level (per channel) {Write}

ThreshMode — Switch between manual and dynamic thresholding (per channel) {Read/Write}

Threshold — Change the threshold level (per channel) {Read/Write}

pNe1 — Filtered plot-decimated data stream {Stream}

eNe1 — Captured spike snippets with assigned sort codes (if Hardware Sort enabled) {Snippets}



Reference

The PCA Spike Sorting gizmo performs filtering, thresholding and online principal component-based spike sorting and storage on multi-channel neural signals at sampling rates up to 50 kHz (up to 100 kHz for very low channel count).

Data Storage

This gizmo generates two types of data for storage: snippet data (includes timestamp, short waveform, and sort code) and plot decimated data streams. The stream data generated by this gizmo is a highly decimated version of the waveforms that keeps local maximum and minimum values of the filtered signals, which makes it ideal for visualizing high frequency spike activity on a computer monitor with a fixed number of pixels.

In plots and in the data tank, each type of data is designated with a prefix: 'e' for snippets and 'p' for streams. You can opt to save only snippets or to disable storage in the gizmo's configuration settings. The sort codes can be configured as an output to be used in other gizmos.

Threshold Detection

At runtime, candidate spikes are detected based on a calculation of the deviation of a waveform from its RMS. By default, the timestamp and position of the waveform in the snippet is dependent on the time of the threshold crossing for the signal. An alternative setting allows waveform timestamp and positioning to be determined by the waveform's highest peak, aligning snippets to their respective peaks. By default, detection is automated and you can make adjustments in the threshold control plot in the runtime window.

Spike Sorting

The sorting interface works in three phases:

1. Training
2. Classification
3. Sorting

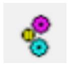
Training

During an initial training period, candidate waveforms are collected and used to compute the first three principal components with the largest possible variance for each recording channel. Incoming waveforms are transformed and appears as dots in the three-dimensional feature space.

Classification

Dots in the feature space are then clustered to isolate waveforms that were recorded from the same neuron. By default, auto-clustering is disabled and no clustering (or sorting) takes place until it's initiated. The default clustering method is a Bayesian algorithm, but you can choose a K-Means method or use manual cluster cutting techniques. Preliminary identification of units is indicated by color coding in the plots provided for visualization; however, all candidate spikes are saved to the data tank with a sort code of 0 during this phase. During this phase you can explore the data and modify sort parameters without affecting saved data.

Sorting



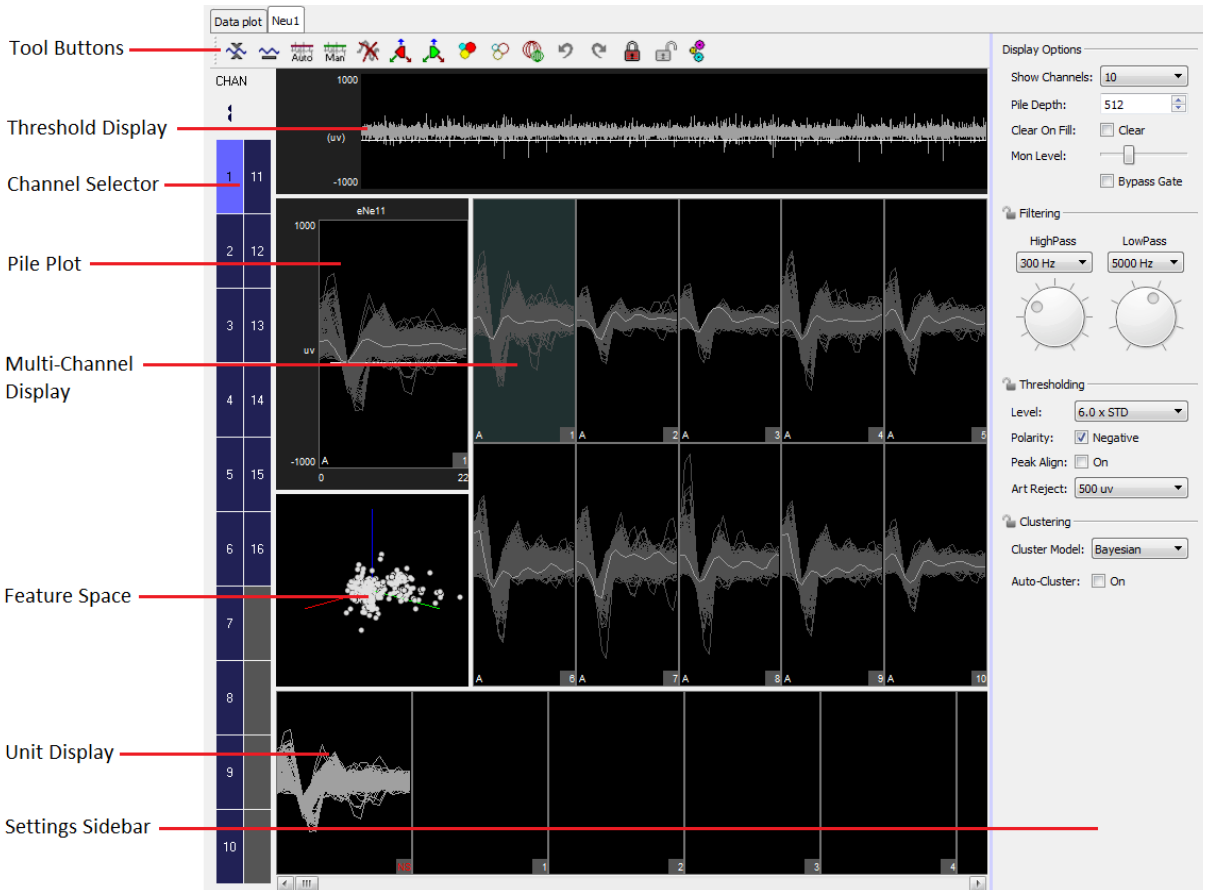
When you're satisfied with the clustering, you can apply **Hardware Sorts**. The clustering parameters are sent to the hardware and sort codes will be applied to new data as it is acquired in real-time. This toolbar button **must** be 'pressed' for online sorting to take place on the hardware.

The Runtime Interface

Runtime Plot

Streamed waveform and snippet plots are added to the runtime window for visualization

PCA Spike Sorting Window



The runtime window includes:

Display	Description
Tool Buttons	Perform actions that are global to all channels.
Threshold Display	Displays the plot decimated waveform of the currently selected channel and the threshold marker. When automatic threshold tracking is active the threshold bar is locked.
Channel Select	Selects the active channel and indicates channel status. Red color indicates training is active and the PCA feature space is being calculated. Blue indicates training is complete. Gray indicates the channel is locked and you can't change sorting parameters.
Pile Plot	Displays candidate spikes for the active channel. Indicators in the bottom left corner denote scaling and threshold tracking states (**A** for automatic, **M** for manual). Users can manually classify waveforms by shape (hold the **Ctrl** key and left-click drag to select waveforms that you want to classify).
Multi-Channel Display	Displays a pile plot for each channel. The channel number is shown in the bottom right corner and new waveforms are highlighted as they are added to the plot. Clicking a subplot makes that channel the active channel for other plots on the tab. Indicators in the bottom left corner denote scaling and threshold tracking states.
Feature Space	Displays the active channel of candidate spikes in three-dimensional PCA space. Manually select waveforms by holding the **Ctrl** key and drawing an arbitrary shape around a visible cluster.
Unit Display	Displays a single channel of candidate waveforms by unit--each plot displays all waveforms classified with a single sort code.
Settings Sidebar	Includes settings for display options, filtering, and threshold settings.

Simple Zoom

You can zoom any plot to see more or less detail without affecting the actual data.

To change the zoom level, hold down the **Shift** key and left-click-drag the mouse up or down.

To reset the zoom level, hold down the **Shift** key and double-click the mouse within the display area.

Display Scale

To make it easier to see waveform shapes for channels with lower magnitude, you can scale individual channels manually or normalize all channels to fit to a similar scale, all without altering the data being stored.

To normalize all channels, click the **Auto Scale** button in the toolbar and choose to normalize the display. Each channel is scaled individually to fit around 80% of the signal's vertical size in each plot. An up or down arrow is displayed in the bottom left corner of the plot or subplot to indicate whether the display has been scaled up or down. This does not change the scale of the feature space.

To adjust the scale of a single channel, press and hold down the **Ctrl** key, and click-and-drag the mouse up or down in the Multi-Channel Display. While adjusting the display scale, the numeric value in the lower right corner of the channel plot indicates the new scale value.

To reset the scale for all channels, click the **Reset Base Scale** button. This does not remove any Zoom applied to a plot.

To return a single channel to its base scale, right-click the desired channel and select **Reset Scaling** from the menu.

Settings Sidebar

Display Options	Description
Show Channels	Select the number of channels to display in the Multi-Channel Display.
Pile Depth	Enter a number to set the maximum number of events displayed in pile plots. The oldest waveform traces are removed as new events are added.
Clear on fill	Select the check box to refresh plots, clearing all traces for a given channel whenever the pile depth is reached on that channel.
Mon Level	Slide the indicator to adjust the level of the audio monitor output, when enabled.
Bypass Gate	A noise gate on the audio monitor removes background noises so only the spikes are heard. Select this check box to turn off the noise gate.

Filtering Options	Description
HighPass/LowPass	Set the highpass and lowpass digital filter settings. The filter is applied to the data before threhsolding, sorting, or visualization

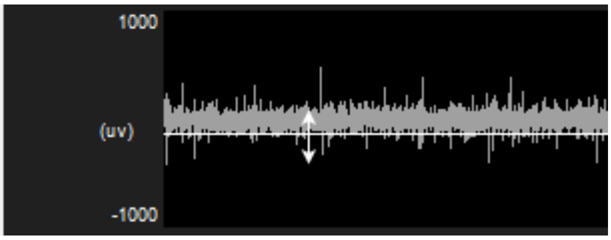
Thresholding Options	Description
Level	Set the automatic threshold level for spike detection, in number of standard deviations from the baseline (smoothed over a 3 second window)
Polarity	Set automatic threshold search polarity, either positive or negative
Peak Align	If enabled, aligns spikes according to their peak values, altering the timestamp of the snippet
Art Reject	When artifact rejection is enabled in the configuration options, sets the artifact rejection level in microvolts. If any sample of the candidate waveform is above this level, the waveform is ignored

Clustering Options	Description
Clustering Model	Select between Bayesian and K-Means sorting algorithms. Bayesian performs automated clustering based on expectation-maximization analysis of Bayesian probabilities. K-Means performs semi-automated clustering using a binary split algorithm that attempts to find the optimum locations of the cluster centers through an iterative process and a defined number of clusters (specified by the **Num Clusters** setting below)
Num Clusters	Set the max number of clusters (2-6) for the K-Means sorting algorithm. If adding another cluster does not improve the efficiency of the algorithm it is not added.
Auto-Cluster On	Select to automatically update clusters for all channels as the feature space is being calculated during training.

Threshold Control

Click the **Auto Threshold** button to initiate automatic threshold tracking on all unlocked channels. If auto thresholding is enabled in the designtime interface, real-time tracking will begin on all channels, otherwise the channels will remain in manual threshold mode and the threshold will be set based on a one-time calculation using the current window data and the **Thresholding Level** and **Polarity** settings.

Click the **Manual Threshold** button to enable manual thresholding on all unlocked channels. In manual threshold mode, the threshold bar may be adjusted by clicking and dragging the white bar in the threshold display window (shown below) or in the pile plot.

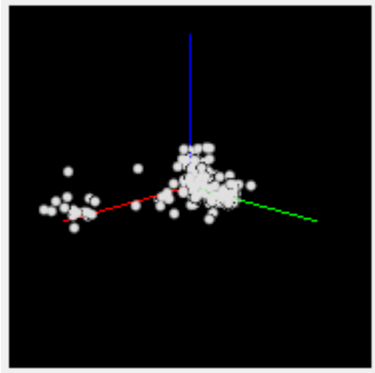


Threshold Display in Manual Mode

You can also right-click the plot at the desired threshold location and choose **Set Threshold Here** from the menu to move the threshold to that location on one channel. You have the option to apply this new location to all channels in manual thresholding mode.

Right-click the pile plot or threshold display and use the **Auto/Manual Threshold** options to change the threshold mode of an individual channel.

Feature Space



Feature Space

Viewing Events in the Feature Space

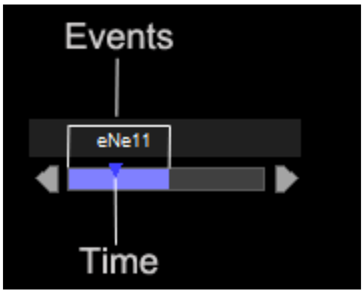
Click-and-drag in the feature space to rotate the view.

Press and hold the **Alt** key on the keyboard and click-and-drag to pan the feature space.

Press and hold the **Shift** key on the keyboard and double-click anywhere in the feature space to reset the feature space view.

Training

During training, as events are added to the display they become part of the data set used for feature space calculation. The feature space for each channel is periodically recalculated using all data in the history at that time, until the training is complete. Training ends when either the required number of events is reached or the training time period expires.



Training Bar

During the training period, a colored progress bar (shown above) is added to all pile plots to show how many events are required, or how much time has elapsed. By default, the progress bar is colored blue. If Auto-Cluster is enabled in the settings sidebar, the progress bar is red.

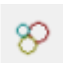
Arrows located on either end of the training progress bar can be used to restart the training period (left arrow) or to accept the current feature space (right arrow) for the active channel.

Click the **Accept Current Space** button to accept the current feature space for all channels. Accept the feature space on individual channels by right- clicking on any plot of an actively training channel and selecting **Accept Space**.

Training on all channel can be initiated by clicking the **Recalculate Space** button. Training can be initiated on individual channels by right-clicking any plot and selecting **Recalculate Space**.

Using Clusters for Classification

Click the **Cluster Automatically** button to calculate clusters for all channels using the options in the settings sidebar. If training is active, this stops training and accepts the feature space before clustering. Each waveform identified by a sort code is represented by a single color on all plots. To cluster an individual channel, right-click on the pile plot or threshold display and choose **Auto Cluster**.



Click the **Clear Clusters** button to remove all clusters on unlocked channels. To clear clusters from an individual channel, right-click on the channel plot and choose **Clear Clusters**. Sort codes already saved to disk remain unchanged.

Click the **Show Spheres** button to view the boundaries of spheres used to define cluster shapes in the feature space.

Applying Sorts to New Data

Sort codes are not saved to the data tank until you apply sorts. You can re-sort or make adjustments as needed to get the best results.

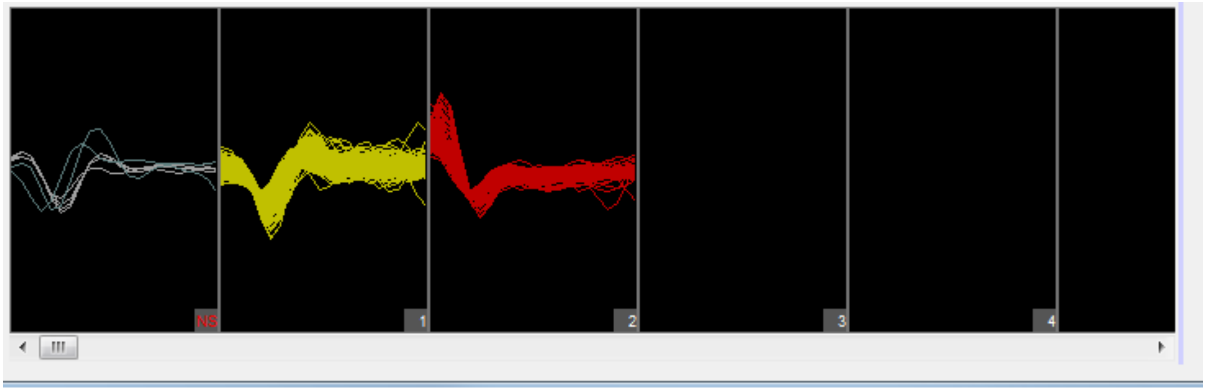
Click the **Hardware Sort** button to send the sorting parameters to the hardware and begin saving sort codes to the data tank. Sort codes are applied as new data is acquired. While this button is down, any changes in sorting parameters in the display will be sent to the hardware and applied automatically to new data.

Locking Channels

Click the **Lock All** button to lock the clusters for all channels, or right- click individual channels and choose Lock . If training is active, locking any channel also ends any the training and accepts the feature space.

Click the **Unlock** button to unlock all channels, or right-click individual channel plots and choose **Unlock**.

The Unit Display



Unit Display

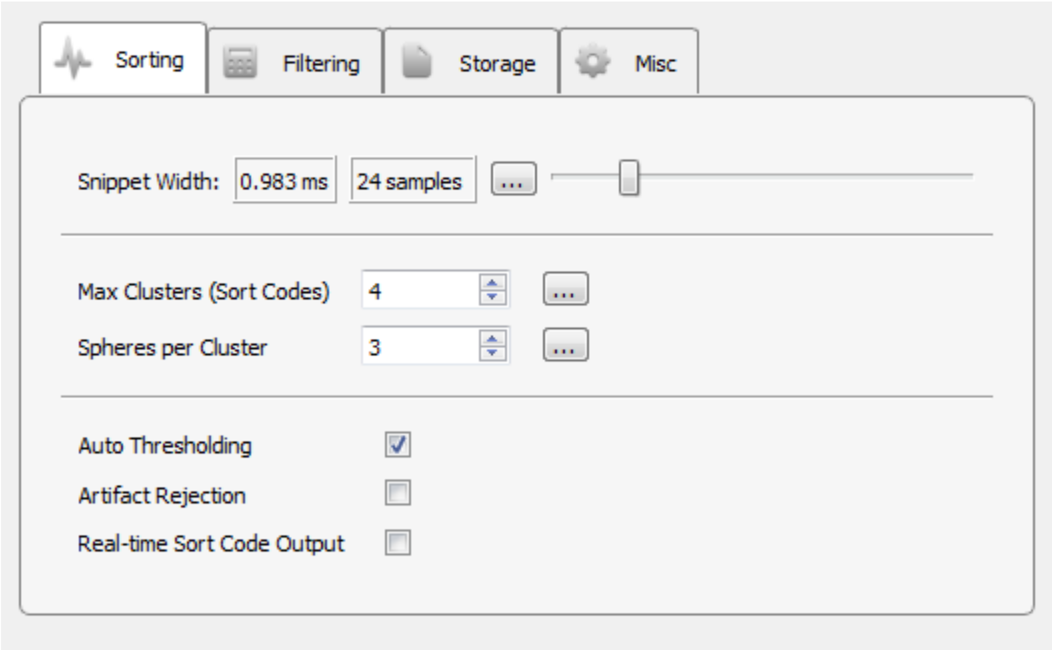
In the Unit Display, candidate waveforms from the currently selected channel are grouped by sort code. Unsorted (sort code 0) and outlier (sort code 31) waveforms are displayed to the left with the label NS.

The maximum number of sort codes (up to five) that can be sorted on the hardware is determined by the **Max Clusters (Sort Codes)** configuration setting. Assigned sort codes larger than this value are displayed in red to indicate they are only visible in the software interface. These waveforms will be given a sort code of 31 (outlier) in the data tank.

The Unit Display can be used to reassign units to different sort codes or combine two or more units together into a single unit by dragging the units.

PCA Spike Sorting Configuration Options

Sorting Tab



Sorting Options Tab

Snippet Width

Drag slider to select the desired width (displayed in milliseconds and samples) of recorded snippets.

Max Clusters (Sort Codes)

Events that contain similar characteristics are grouped into clusters and are given the same sort code. The maximum number of clusters supported in hardware sorting is five. Allowing a larger number of clusters increases processing overhead, but accommodates greater variability in the data set.

Spheres per Cluster

Spheres in the three-dimensional PCA feature space are used to define each cluster. The maximum number of spheres supported is five, per cluster per channel. Allowing a larger number of spheres to the sorting algorithm increases processing overhead, but provides a more accurate fit for a cluster's shape.

Auto Thresholding

In automatic thresholding, the threshold used to record snippets is adjusted in real-time to changes in each channel waveform's RMS. The previous five seconds of data are used in the RMS calculation.

Artifact Rejection

When artifact rejection is enabled, snippets that contain at least one sample greater than the artifact rejection level set on the runtime interface are ignored.

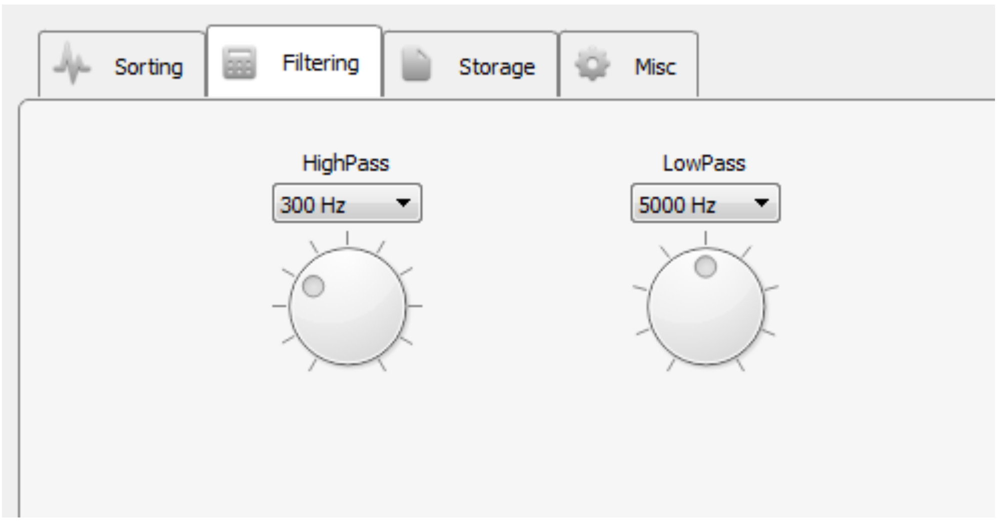
Real-time Sort Code Output

Make the multi-channel integer stream of compressed sort codes available to other gizmos, such as Sort Binner or UDP output.

Note: The sort code output is delayed by (window width + 2) samples from when the threshold is crossed. When artifact rejection is enabled, the sort code output is delayed by an additional window width, so (2 * window width + 2) total samples.

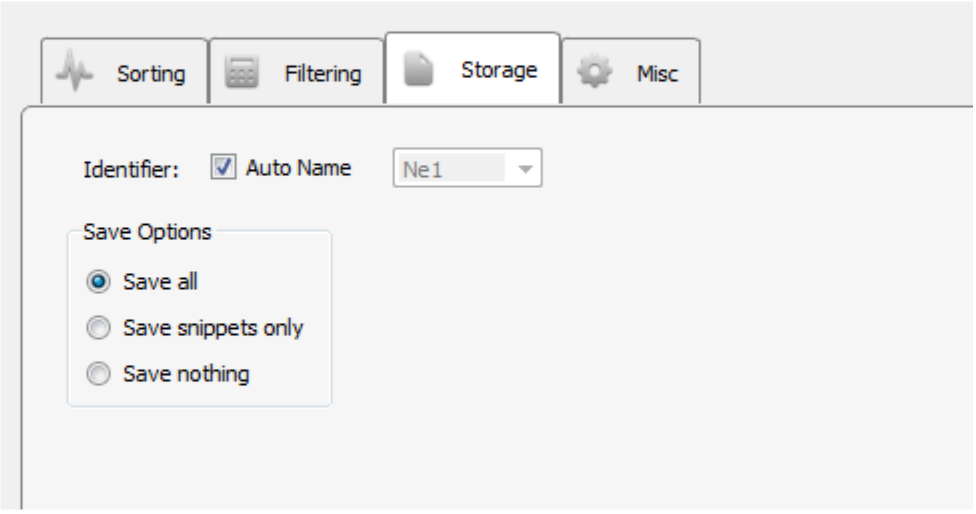
Filtering Tab

The gizmo applies a highpass and lowpass filter to all channels before spike detection. The runtime interface includes controls for dynamic adjustments to the filter settings. You also set default values in the Filtering Options tab.



Filtering Options Tab

Storage Tab

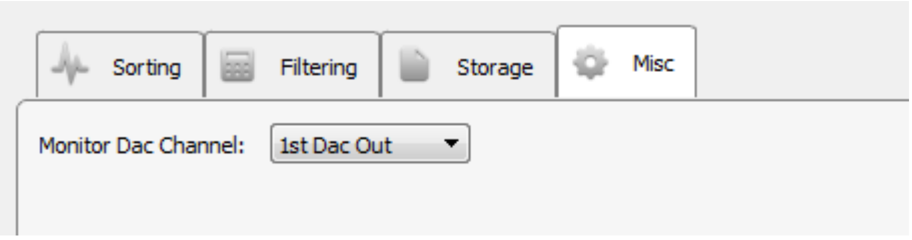


Storage Options Tab

Save Options

Select whether to save only snippet waveforms or to include the plot decimated waveforms used by the sorting gizmo or to save nothing at all to the data tank.

Misc Tab



Misc Options Tab

Monitor DAC Channel

Select an output channel to send the monitor signal to, or set to **Disable** to turn monitoring off.

Pulse Generator

Common Use Cases

Creates user-defined pulse trains based on milliseconds or Hz. Control duty cycle, period, number of pulses, and trigger pulse trains internally or from other gizmos. Use this gizmo for directly controlling optogenetic stimulation or driving the timing of other connected gizmos or devices.

Data Stored	
Epoch event (optional)	Store each pulse as onset/offset epoch events
Continuous (optional)	Store the pulse stream continuously
Outputs	
StrobeOut	Single channel logic, the pulse train
FloatOut (optional)	Single channel floating point, a scaled pulse train signal

Gizmo Help Slides

Quick Help Slide 1

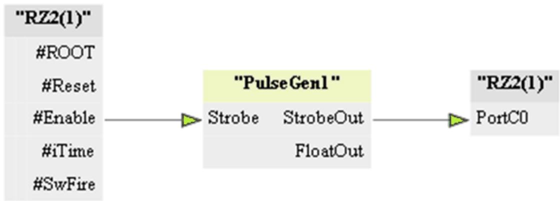
Pulse Generator — Create user-defined pulse trains in milliseconds or Hz

Optogenetic stimulation

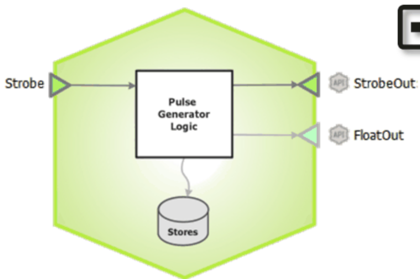
External device triggering

Third party device synchronization

Sending a TTL pulse train to the first bit out on an RZ2



Strobe: Logic strobe signal from other gizmos or digital I/O



StrobeOut: Output TTL logic signal of pulse train

FloatOut: Output floating-point signal of pulse train

Quick Help Slide 2

Pulse Generator

User Interface, API, Data Stored

Adjust the parameters of individual pulses and the overall pulse train

DutyCycle — Percentage or time of Pulse that is logic high {Read/Write}

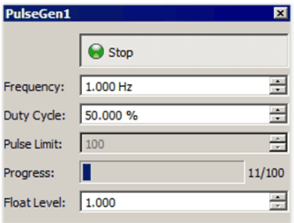
Enable — Trigger pulse generator {Read/Write}

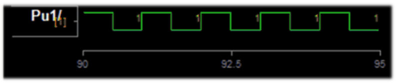
PulseLimit — Total number of pulses a pulse train will run {Read/Write}

PulsePeriod — Time between onsets of consecutive pulses {Read/Write}

PulsesCompleted — Count of the number of presented pulses {Read}

FloatOutValue — If enabled, floating point output value of this {Read/Write}





Pu1/ — Store the pulse output with onset/ offset timestamps {Epoch}

Continuous — Store the pulse train as a continuous waveform {Epoch}

Reference

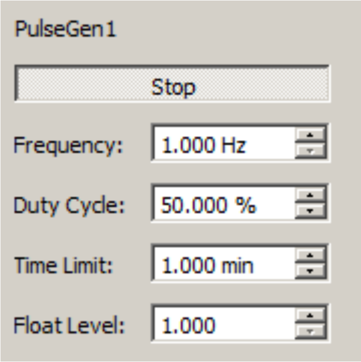
The Runtime Interface

Runtime Plot

At runtime, the standard Synapse data plot is available to display any stored data. The pulse generator can store its output waveform as onset/offset epoch events and/or as a continuous stream.

Pulse Generator Tab

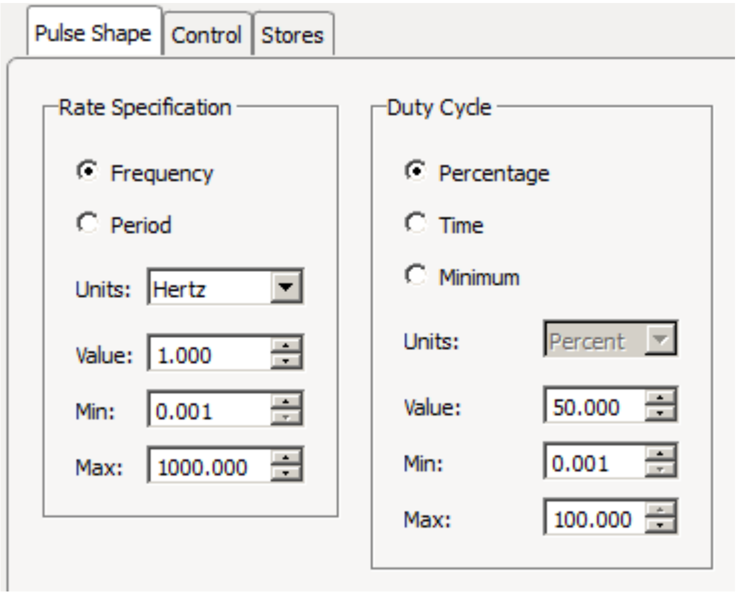
When **Enable Run-time Interface** is selected, the user can dynamically control the pulse timing and pulse width, as well as enable/disable pulse output in some modes.



Pulse Generator Runtime UI

Pulse Generator Configuration Options

Pulse Shape Tab



Pulse Shape Tab

Rate Specification

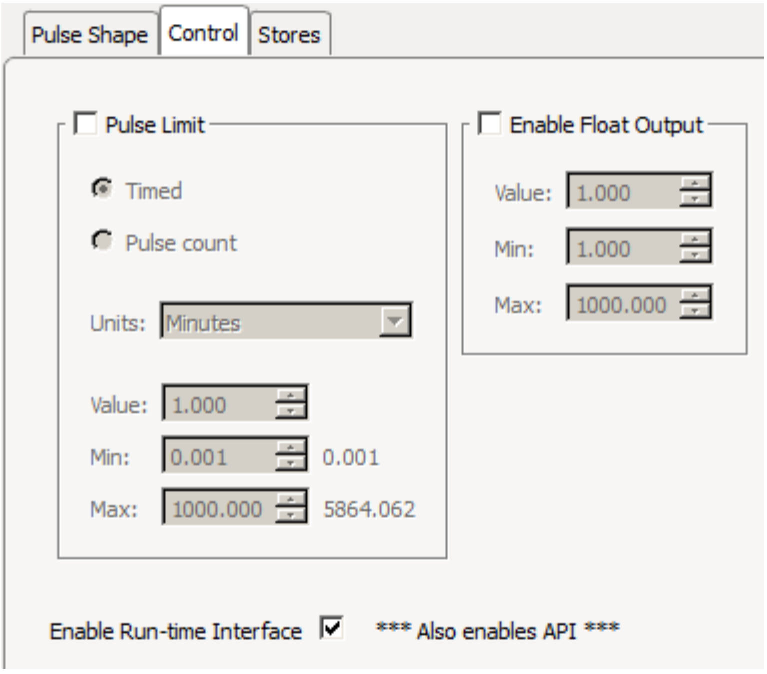
Choose whether the pulse rate is defined by Frequency or pulse Period.

Frequency defines the onset to onset time in units of hertz, kilohertz, or beats per minute (BPM). **Period** defines the onset to onset time in minutes, seconds, milliseconds, microseconds, or samples. The default Value as well as bounds on the Min and Max values are set.

Duty Cycle

Choose the pulse high time as either a Percentage of the onset to onset time, or by a fixed Time duration, in units of minutes, seconds, milliseconds, microseconds, or samples. Select Minimum for single sample pulse width.

Control Tab



Control Tab

Pulse Limit

Determine when to activate the pulse generator. If this box is unchecked, the pulse generator is active as long as the Enable gizmo input is active or the user has pressed the 'Enable' button on the runtime interface (Run-time Interface must be enabled for the button to appear).

Timed means the pulse generator runs for the specified duration whenever it is triggered.

Pulse Count means the pulse generator runs for the specified number of pulses. It can be stopped by clicking the 'Stop' button on the runtime interface.

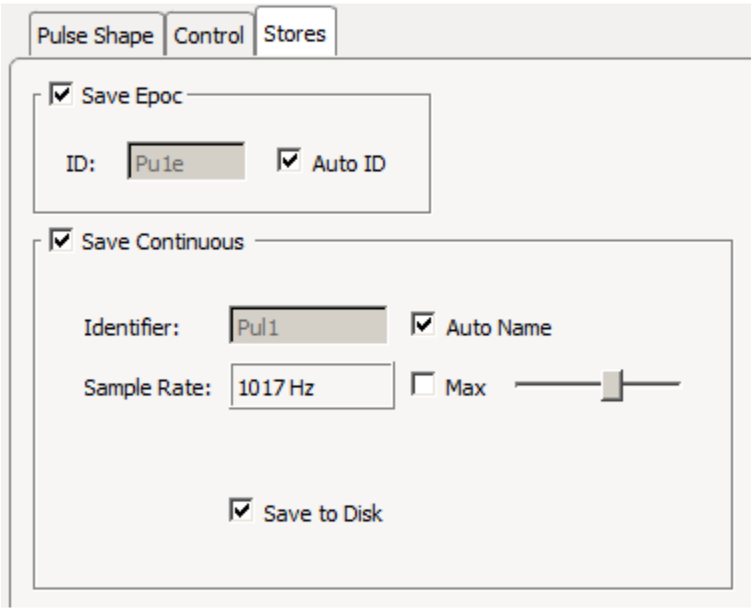
Float Value Output

If enabled, adds a secondary floating point gizmo output with the user-defined scale factor.

If **Enable Run-time Interface** is checked, all Control and Pulse Shape values can be adjusted by the user at run time, subject to the Min/Max bounds specified at design time.

In Timed and Pulse Count modes, the user interface button changes to 'Start/Stop' and only the rising edge of the input is used to trigger the pulse train. The user has the ability to end the pulses before they are finished by disabling the runtime interface button. If the pulse train is initiated by the gizmo input, the button detects this and changes to 'Stop', letting you prematurely halt the pulse train by clicking the button.

Stores Tab



Stores Tab

Choose what data (if any) to store.

Save Epoc

Store the timestamp of the pulse onset/offset events.

Save Continuous

Saves the pulse train continuously. If the floating point output is enabled, this is the value that is stored.

Pulse Train Generator

Common Use Cases

Create simple or complex user-defined pulse train waveforms based on a number of configuration options that include stacked or parallel trains. Control whether trains are strobed or triggered, and define waveform characteristics with a parameter table. Use this gizmo for directly controlling optogenetic stimulation or other connected gizmos or devices. Pulse Train is especially useful when you want to create complex pulse trains where one signal is gating another, or if you want to inform waveform parameters dynamically from other gizmos.

Data Stored	
Epoch event (optional)	Store each pulse as onset/offset epoch events
Epoch event (optional)	Parameter values when triggered
Continuous (optional)	Store the pulse stream continuously
Outputs	
Train-{n}	Single channel logic, integer, or floating-point value
Active	Logic true while train is active

Gizmo Help Slides

Quick Help Slide 1

Pulse Train Generator — Create simple or complex user-defined pulse train waveforms based on several configuration options that include stacked or parallel trains.

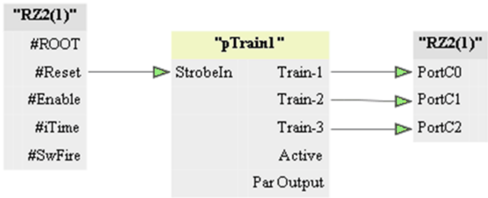
Optogenetic stimulation

External device triggering

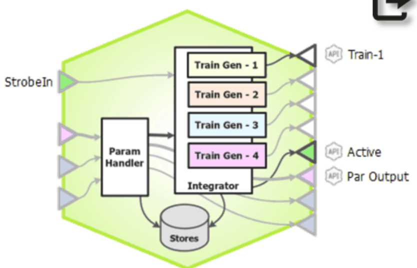
Third party device synchronization

Dynamic pulse parameter modulation

Sending TTL pulse trains to the first three output bits on an RZ2



Strobe: Logic strobe signal from other gizmos or digital I/O. Can be used to trigger train or enable/disable



Train-{x}: Output TTL logic signal, integer or floating-point value, of pulse train. The Train-{x} output type is set in Generators → Output

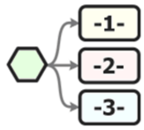
Active: TTL logic signal when pulse train is active

Quick Help Slide 2

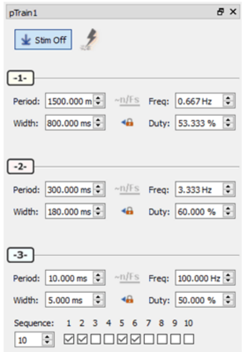
Pulse Train Generator

User Interface, API, Data Stored

Configuration



Adjust the parameters of individual pulses and the overall pulse train



Amplitude{1...n} — Height of waveform when output set to float or interger {Read/Write}

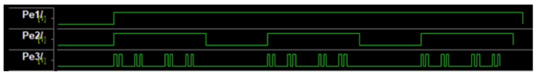
Count{1...n} — Number of pulses in a triggered train {Read/Write}

Mute — Mute output of train {Read/Write}

Period{1...n} — Onset to onset period of train pulses {Read/Write}

Stim Active — TTL signal high when pulse is active {Read}

Strobe — Trigger or toggle train on/ off {Read/Write}



On Stim — Store onset/offset times for duration of a stim bursts in a train {Epoch}

On Train{n} — Store onset/ offset times of each pulse in a burst {Epoch}

Quick Help Slide 3

Pulse Train Generator

Parameter Table

This gizmo has a Parameter Table to dynamically control values its parameters from other gizmos and during runtime

Parameter Tables provide the user dynamic control over parameter values. Each one of the Modes will change how the user can control and interact with the parameter.

Param In – Dynamically control parameter values from Parameter Sequencer or Parameter Manifold

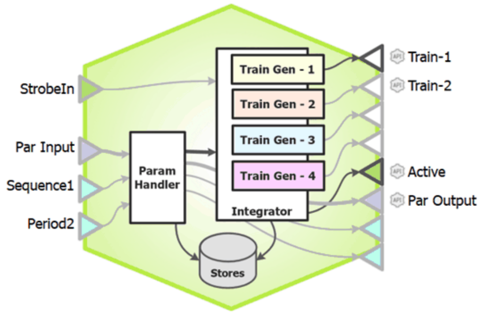
Constant – The value is immutable at runtime

Scalar In – Pass a scalar value from another gizmo for dynamic real-time control

Widget – Creates a widget (slider) on the user interface that is controlled manually by the user or through the API.

Scalar Out (Scout) – Makes this value available as a gizmo output to connect to another gizmo input in real-time

Epoc – Save the parameter values used for each stimulus presentation



Changing the Mode of parameters will change the input and output options on the gizmo.

All parameters can be read through API. Only parameters in Widget mode can be written to.

All parameters are connected to the “Par Output” gizmo output link

Parameters:

	Name	Mode	Value	Jit(%)	Min	Max	Epoc	ID	Auto ID	SCout-1	SCout-2
1	Period1 (ms)	Constant	75.000	0.0	0.100	100000.0(	on Stim	Pe1/	☑	○	○
2	Width1 (ms)	Widget	10.000	0.0	0.100	100000.0(	on Train1	Wi1/	☑	○	○
3	Count1	Param In	10	0.0	1	10000000(	None	Cnt1	☑	○	○
4	Sequence1	Scalar In-1	0	0.0	0	10000000(	None	Seq1	☑	○	○
7	Period2 (ms)	Scalar In-2	100.000	0.0	0.100	100000.0(	None	Per2	☑	○	○
8	Width2 (ms)	Constant	10.000	0.0	0.100	100000.0(	None	Wid2	☑	○	○
9	Count2	Widget	1	0.0	1	10000000(	None	Cnt2	☑	○	○
11	Delay2 (ms)	Param In	0.000	0.0	0.000	10000000(	None	Del2	☑	○	○

Reference

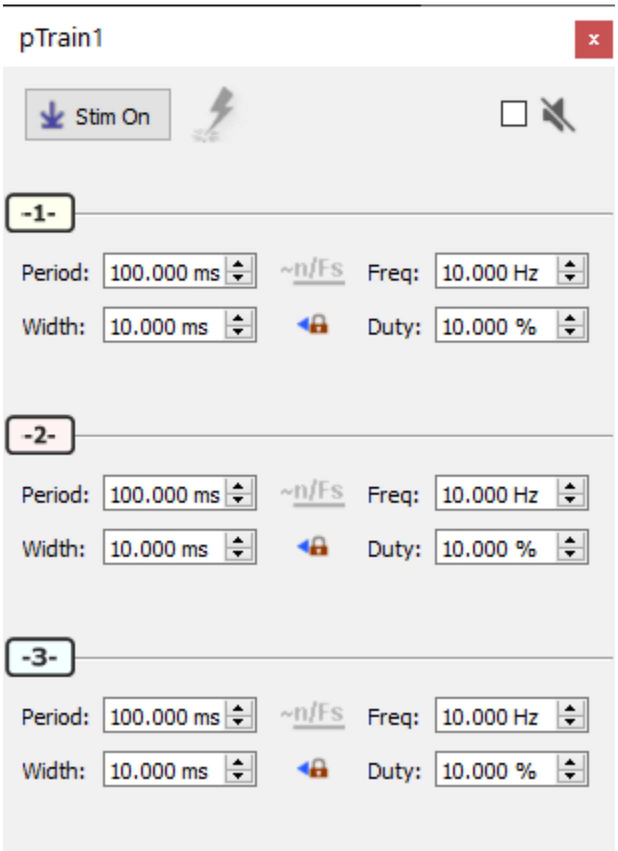
Pulse Train Generator creates a user-defined sequence of pulses. Creates up to 4 different pulse sequences that can be ANDed together.

The Runtime Interface

Runtime Plot

At runtime, the standard Synapse data plot is available to display any stored data. The pulse train generator can store its output waveform as onset/offset epoch events.

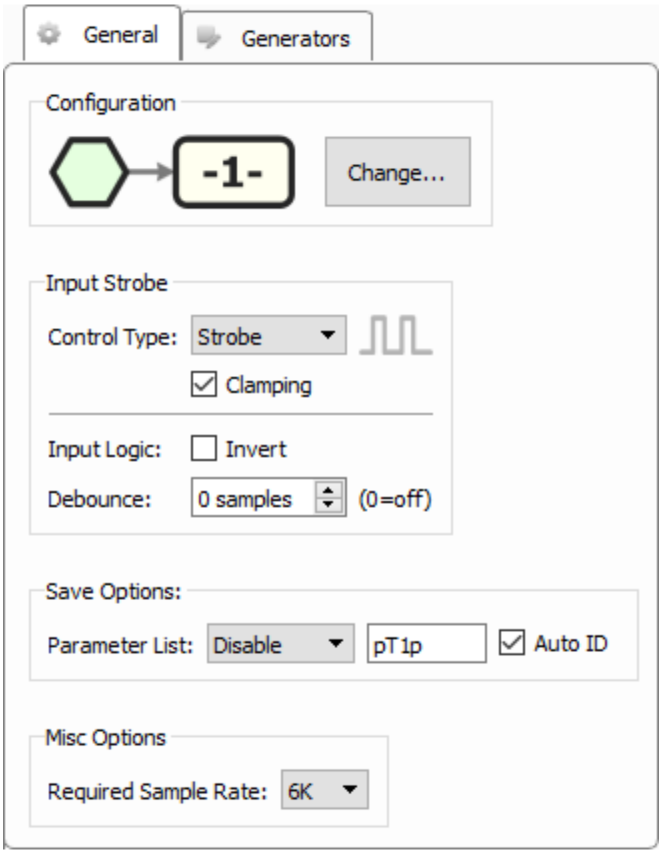
The user can dynamically control any parameter set to Widget mode (the default) at run-time, as well as enable/disable pulse output and Mute the output.



Pulse Train Generator Runtime UI

Pulse Train Generator Configuration Options

General Tab



There can be up to 4 trains synchronized together in one Pulse Train Generator gizmo. Use the **Change...** button to select the configuration. See Generator Order Options below.

Input Strobe

Determines how the master Enable input into the gizmo controls the train presentation.

Control Type **Triggered** means the trains begin on the rising edge of the gizmo input.

Control Type **Strobe** means the trains are enabled while the gizmo input is high.

When Control Type is set to Strobe, **Clamping** means the train is immediately turned off when the gizmo input goes low. If Clamping is disabled, the train pulse is allowed to finish.

Invert Input Logic and **Debounce** are typically used if the input is coming directly from a digital input on the RZ processor.

Parameter List

Select whether to store the value of all parameters and on which train onset to store them. This generates a multi-channel list of scalar values. The channels map directly to the rows of the parameters table on the Parameters tab. By default, some parameters are hidden in the table, but values are stored for all parameters.

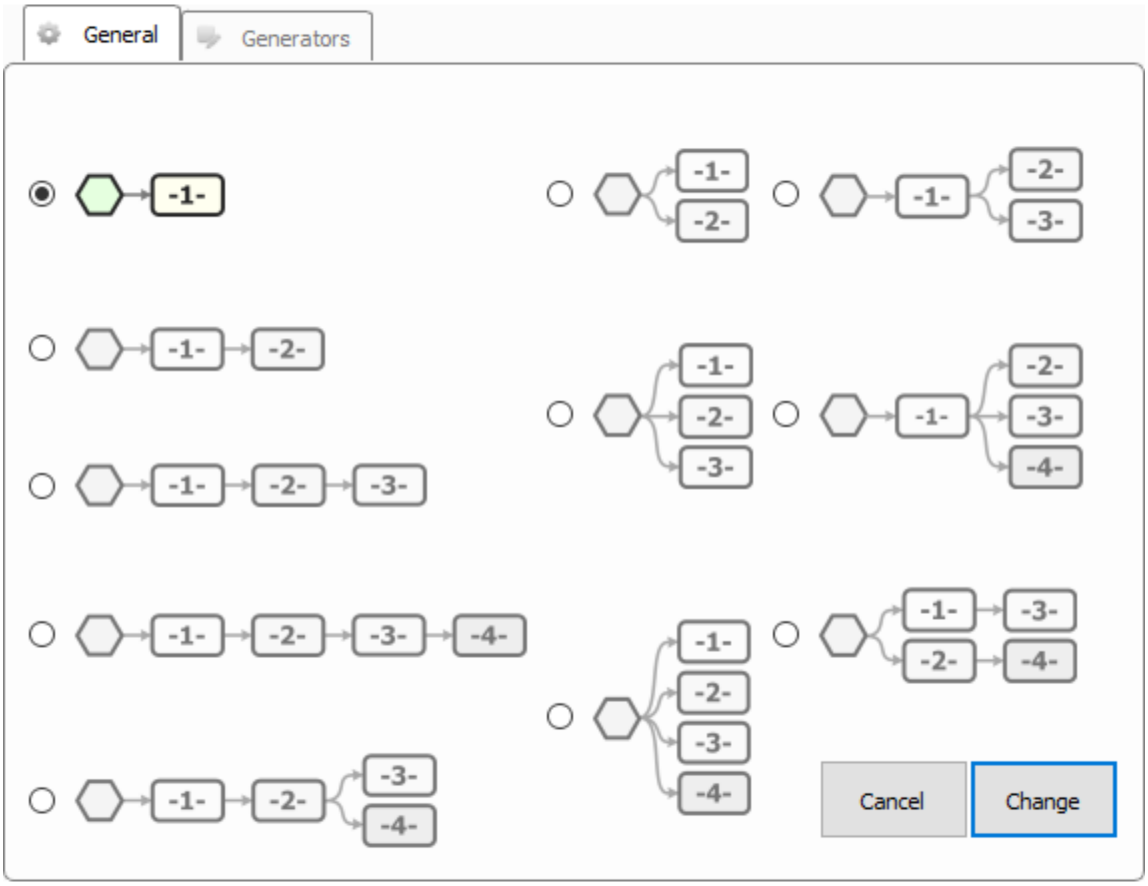
Auto ID field and check box

A store name is generated automatically. A "/" is appended to the name to indicate when the full epoc is stored (and is not appended when only saving the onset). To use your own store name, clear the **Auto ID** check box.

Required Sample Rate

The minimum rate required. Synapse looks through the entire experiment and your rig and sets the sample rate according to this and other limiting factors.

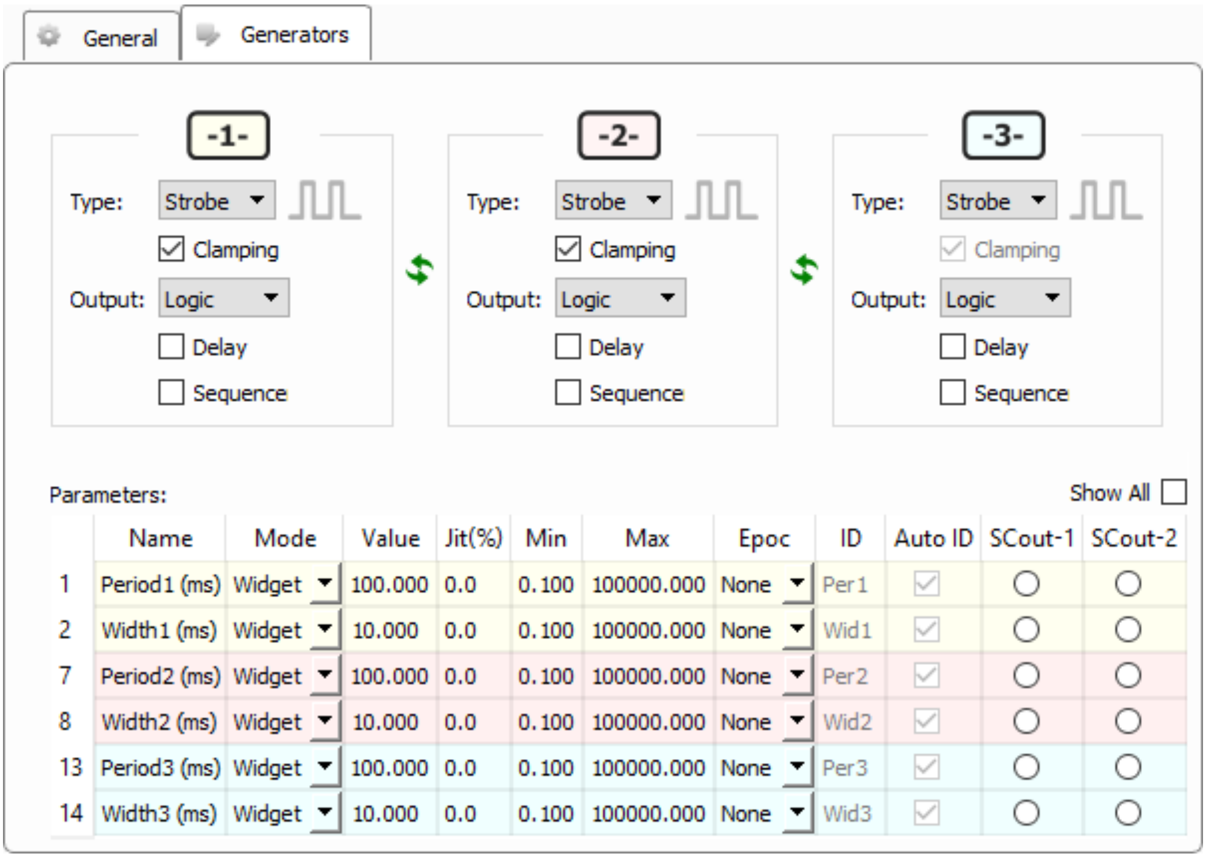
Generator Order Options



Generator Order Options

Choose the number of pulse train generators and the control order based on your desired application.

Generators Tab



Each parent generator has its own set of parameters for how it controls its child generator.

Type **Triggered** means the child generators begin on the rising edge of this generator.

Type **Strobe** means the child generators are enabled while this generator is high.

Type **Timer** generates a single pulse of the desired Width at the desired Delay.

When Type is set to Strobe or Timer, **Clamping** means the child generators are immediately turned off when this generator goes low. If Clamping is disabled, the child generators are allowed to finish their current pulse.

Output can be a Logic, Float, or Integer (Float and Integer values are controlled by the Amplitude parameter).

Delay lets you delay the pulse timing with the Delay parameter.

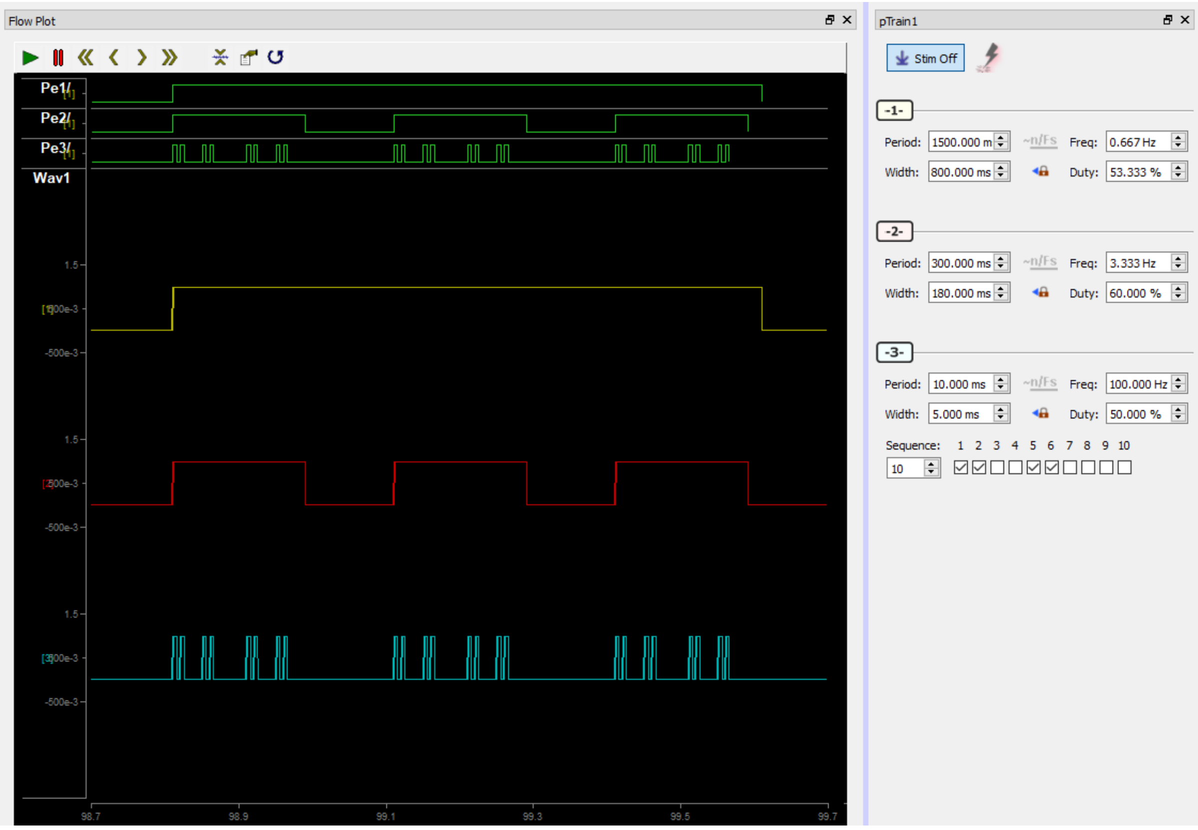
Sequence adds a run-time display that lets you control ordering of presentation. Use this if you want to play a more complex stimulation pattern. See Runtime Sequencing below.

Period defines the onset to onset time in ms. The default Value as well as bounds on the Min and Max values are set.

Width defines the pulse high time in ms

Parameters can be stored as epoc events on the master train or on individual trains.

Runtime Sequencing



Runtime Sequencing Example

In this example, three generators are stacked. Generator 1 is the parent of Generator 2 who's child is Generator 3. For the on duration of a Generator 1 pulse, Generator 2 will pulse with on/off times defined by its period and width. The same is true for Generator 2 and 3. Importantly, the Generator with the largest width and period is at the very top.

Generator 3 uses the Sequence option and only presents on the first, second, fifth, and sixth trigger of its timer.

If using SynapseAPI or Parameter Sequencer to control this playback sequence, the structure of the Sequence{N} parameter is an integer where the upper 16 bits indicate the sequence count (up to maximum 16) and the lower 16 bits are the enable code for each step in the sequence. For example, using SynapseAPI in Matlab:

```
syn = SynapseAPI();

s = [1, 1, 0, 0, 1, 1, 0, 0, 0, 0]
v = bitshift(length(s), 16)
for i = 1:length(s)
    if (s(i))
        v = bitset(v, i);
    end
end

syn.setParameterValue('pTrain1', 'Sequence3', v);
```

Python Coding Gizmo

Common Use Cases

The Python Coding Gizmo ("Pynapse") tightly integrates Python coding into your Synapse experiment. With Pynapse you can: control your experiment flow, build and deliver stimuli, run complex behavioral paradigms, do custom analysis and visualization.

Data Stored	
Epoc (optional)	Timestamps for each input, output, and state change
Outputs	
Logic (optional)	Control signals
Continuous (optional)	Custom stim waveform
Parameters (optional)	Control of stim parameters

Reference

See the full Pynapse User Guide here.

Video Demonstrations

Follow along with these videos for a full experiment walkthrough using Pynapse.

Intro to iCon & Pynapse: Part 1

Learn how to set up the iCon in Synapse and use the Pynapse gizmo for behavioral experiments.

Intro to iCon & Pynapse: Part 2

Learn how to use the Pynapse State Machine to create scalable behavioral experiments.

Selector

Common Use Cases

Pick off individual channels from a multi-channel stream, or isolate specific channel and sort code combinations from Sort Binner. Use this gizmo for routing individual channels for monitoring or further processing, or for reading Sort Binner outputs of single channel + sort code information.

Data Stored	
Stream (optional)	Selected channels
Stream (optional)	Selected channel numbers
Outputs	
Selected channels	Up to four selected channels

Gizmo Help Slides

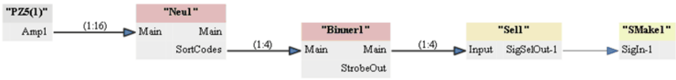
Quick Help Slide 1

Selector — Pick off individual channels from a multi-channel stream, or isolate specific channel and sort code combinations from Sort Binner

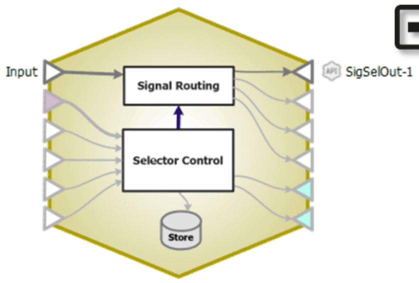
Count the number of spikes that occur in a time window

Send individual LFP and EEG channels to a sleep state detector

Multi-channel spike binning to test the number of sort codes that appear on specific channels within a time-based window



Main: Input fed from neural amplifier, Sort Binner, or any other multi-channel stream



SigSelOut:* Isolated channel (or sort code from Sort Binner) for further processing. If Logic Output is selected, output is a logic 1 or 0.

Quick Help Slide 2

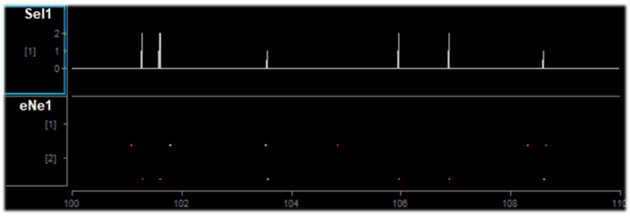
Selector

User Interface, API, Data Stored

Dynamic control of selected channel and in some cases selected sort code

*ChanSel-** — Pick the channel selector monitors {Read/Write}

*SortSel-** — Pick the sort code to monitor on the selected channel {Read/Write}



Bitfields — Save the sort code that occurred on the selected channel {Stream}

Sel1 — Save waveform of selected channel {Stream}

Quick Help Slide 3

Selector

Parameter Table

This gizmo has a Parameter Table to dynamically control values its parameters from other gizmos and during runtime

Parameter Tables provide the user dynamic control over parameter values. Each one of the Modes will change how the user can control and interact with the parameter.

Param In – Dynamically control parameter values from Parameter Sequencer or Parameter Manifold

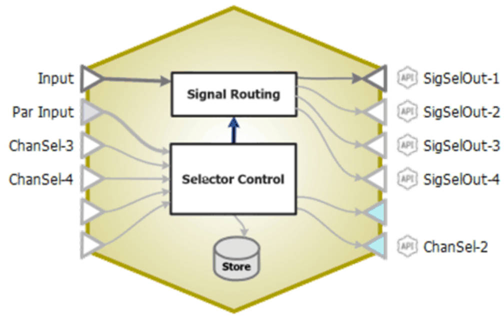
Constant – The value is immutable at runtime

Scalar In – Pass a scalar value from another gizmo for dynamic real-time control

Widget – Creates a widget (slider) on the user interface that is controlled manually by the user or through the API.

Scalar Out (Scout) – Makes this value available as a gizmo output to connect to another gizmo input in real-time

Epoc – Save the parameter values used for each stimulus presentation



Changing the Mode of parameters will change the input and output options on the gizmo.

All parameters can be read through API. Only parameters in Widget mode can be written to.

All parameters are connected to the “Par Output” gizmo output link

General | Parameters

Selection Parameters: Show All

	Name	Mode	Value	Jit(%)	Min	Max	Epoc	ID	Auto ID	SCout-1
1	ChanSel-1	Widget	1	0.0	1	16	None	Chn1	☑	◉
3	ChanSel-2	Param In	2	0.0	1	16	None	Chn2	☑	○
5	ChanSel-3	Scalar In-1	3	0.0	1	16	None	Chn3	☑	○
7	ChanSel-4	Scalar In-2	4	0.0	1	16	None	Chn4	☑	○

Reference

The Selector converts a multi-channel stream into individual channels that can then connect to other gizmos. Integer streams can be further sub-divided to access portions of compressed data, like sort codes from the spike sorting gizmos or the compressed sort code output from the Sort Binner gizmo. Selection can be controlled dynamically through a runtime slider or a gizmo input. Output channels and channel selections can optionally be saved.

Selector Configuration Options

General Tab

The Selector handles multi-channel streams of floating point or 32-bit integer values. By default, you choose individual channels from the multi-channel stream to send to the outputs, depending on the settings in the Parameters tab.

Selection Options

The Selection Options are only available if the Main input to Selector is a multi-channel integer stream. You tell Selector how the data is packed into the 32-bit integers and it will properly extract it, otherwise it will automatically set these values.

Bit Fields

If the incoming data has been compressed, use the **Bit Fields** option and indicate how many bits per channel you used in **Bits Per Field**.

For example, suppose you pack sixteen 8-bit integers into four 32-bit integer channels and send it to the RZ UDP interface, and connect the UDP component to Selector. The Main input into Selector will see a four channel stream of 32-bit integers. Set the **Bits Per Field** to Eight Bits and you can extract channels 1-16 on the output side.

You can also extract a particular channel of sort codes from any of the spike sorting gizmos with this.

Sort Codes

If you have multiple sub-fields for each channel, use the **Sort Codes** option to indicate how many sub-fields you have (**Number of Sort Codes**) and how many bits are in each sub-field (**Bits Per Sort Code**). The most common use of this is to extract a particular channel/sort code count from the output of the Sort Binner gizmo to drive real-time decision making in other gizmos (e.g. State Maker).

When you connect a Sort Binner output to the Selector, Synapse automatically sets the **Bits Per Sort Code** and **Number of Sort Codes** based on the settings in Sort Binner and updates them automatically for you if they are changed in the parent Sort Binner gizmo. If a different type of gizmo is generating the multi- channel integer data (e.g. UDP gizmo or user gizmo), these settings can be defined manually.

This option adds additional rows to the parameters table so you can define the channel and the sub-field you want to extract.

Logical Outputs

When selected in **Bit Fields** mode, output is a logic 1 if selected field's value equals selected sort code value.

When selected in **Sort Codes** mode, output is a logic 1 if selected field's value matches the target sort code.

Save Options

The four selected output signals can be stored continuously or not at all.

Parameters Tab

Use the parameters table to define how the channels are selected. See Using Parameters for more information on working with parameters tables.

Signal Accumulator

Common Use Cases

Collect a sum of an incoming signal over a user-defined window. Optionally compute the average as well. Can also perform thresholding of accumulated signal for further processing. Use this gizmo to calculate the total power of a signal over a specified time span, or to compute average signal power over many trials.

Data Stored	
Scalars (optional)	Timestamped accumulator output
Epoc (optional)	If thresholding is enabled, stores timestamp and selected channel accumulator output when threshold conditions are met

Outputs	
Main	Accumulator output for all input channels
Active	True while accumulator is running
Done	True when accumulator is finished and new value is ready on output
ThrSel	Accumulator output for selected channel in plot
ThrPass	True if selected channel accumulator value meets threshold criteria

Gizmo Help Slides

Quick Help Slide 1

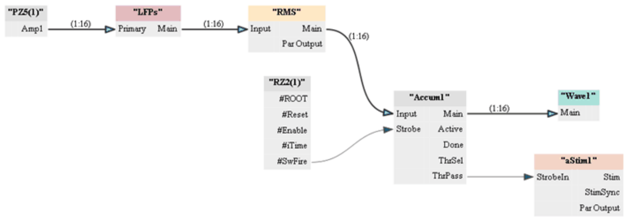
Signal Accumulator — Collect a sum of an incoming signal over a user-defined window with optional average computation and thresholding

Neural recording with action potentials

Acute neurophysiology

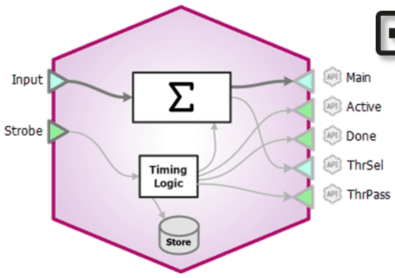
Chronic behavior experiments

Accumulate the total RMS of an LFP signal every second. Present an auditory stimulus if total RMS exceeds a threshold.



Input: Incoming data signal to be analyzed

Strobe: Control timing of accumulator output



Main: Accumulator output signal for all channels

Active: Logic high when accumulator is running

Done: Logic high when accumulator finishes

ThrSel/ThrPass: Selected channel output and logic signal

Quick Help Slide 2

Signal Accumulator

User Interface, API, Data Stored

Dynamic control of selected channel and thresholds.

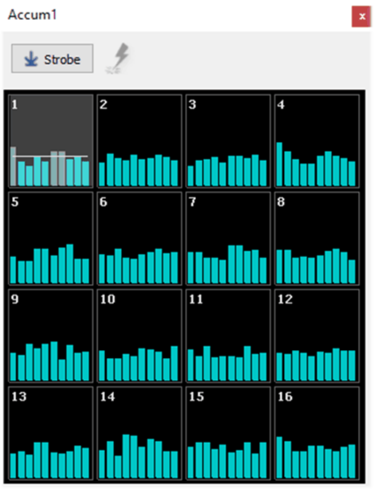
Strobe — Manually initiate an accumulation period {Write}

Channel — Currently selected channel for thresholding {Read/Write}

Threshold — Threshold level to compare selected channel {Read/Write}

Ac1T — Timestamp/value of selected channel when it meets threshold criteria {Epoch}

Acc1 — Timestamped accumulator output for all channels {Strobed}



Reference

The Runtime Interface

Runtime Plot

Scalar and epoch plots, if enabled, are added to the runtime window for visualization when enabled in the gizmo options. See Flow Plot for more information on using and customizing the plots.

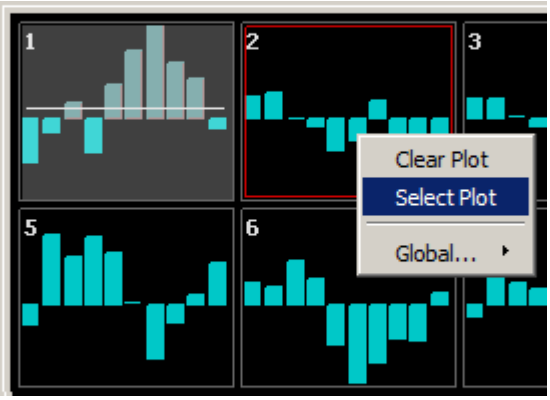
Signal Accumulator Plot

The runtime Signal Accumulator plot must be configured in the Edit mode options before it can be used. See See Plot Preview Tab. for more information.

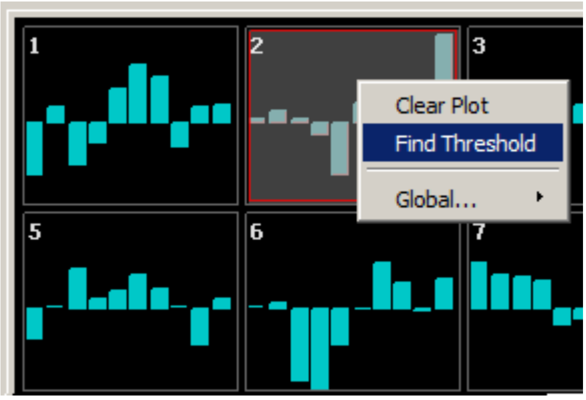
Important

Accumulator is unique in that the runtime changes also modify the experiment setup. Any changes you make to the plot configuration at runtime are saved with the experiment design and vice versa. This goes against the typical Persistence model in Synapse where runtime settings are seperate from designtime settings. This helps you 'set up' the plot you want in the experiment settings using real data in a runtime mode.

Use the right-click menu to choose which channel is actively used with the threshold detector.



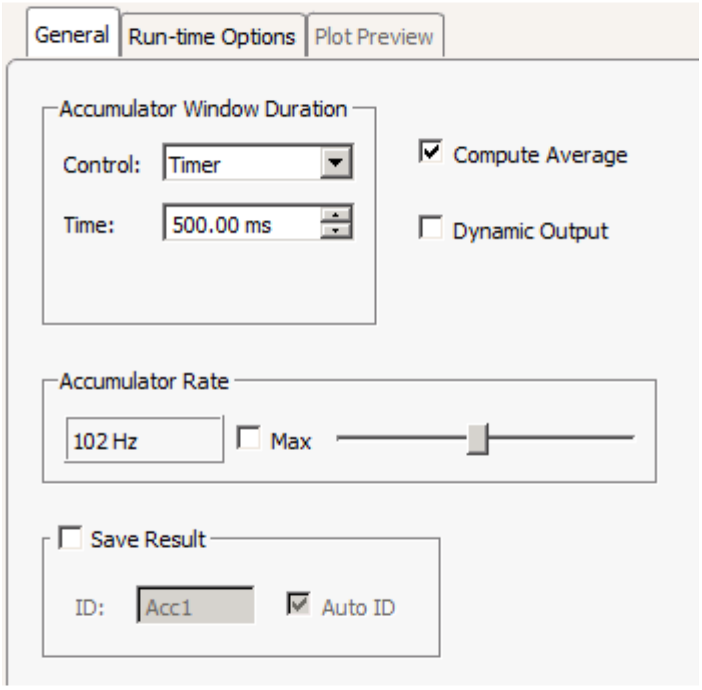
Once the desired plot is selected, if the threshold is not visible then right- click and select Find Threshold.



Once the threshold is visible, use the left mouse button to click-drag the threshold bar into place.

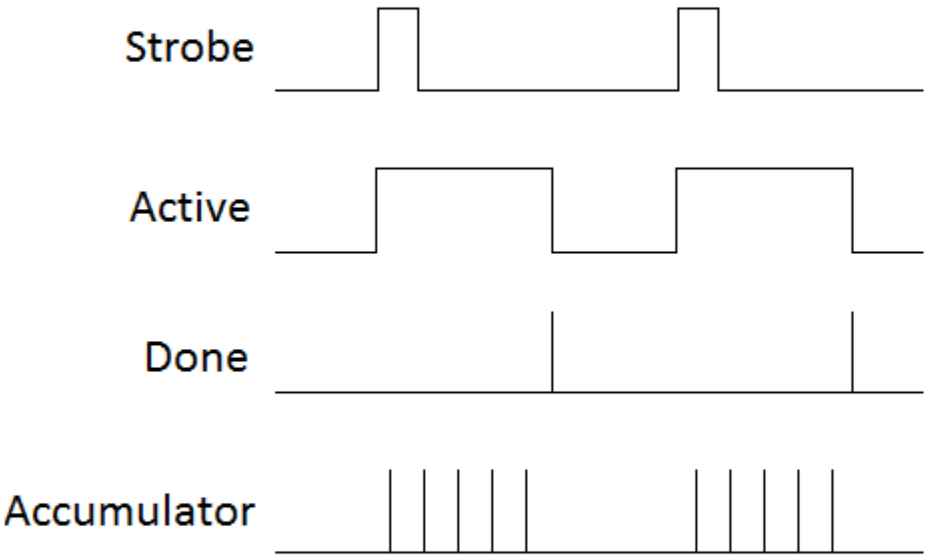
Signal Accumulator Configuration Options

General Tab



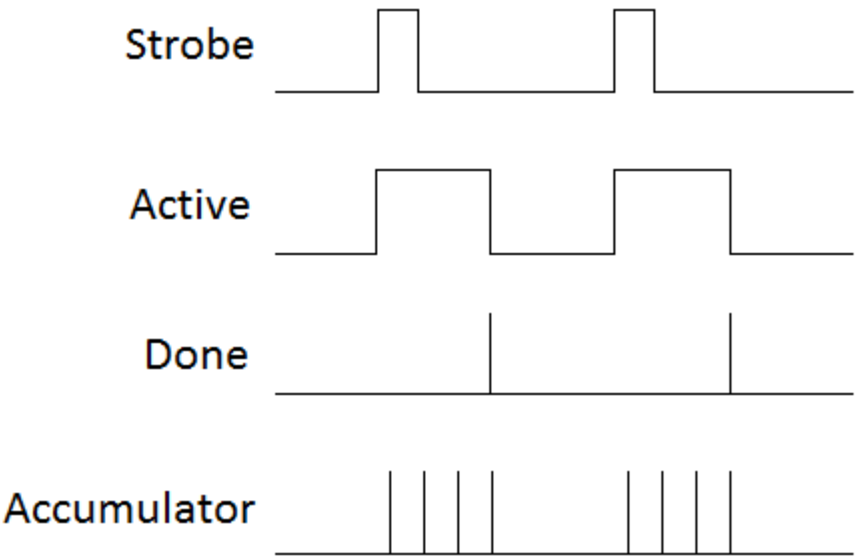
General Tab

The Accumulator samples at the **Accumulator Rate**. The start and end of the accumulation period is controlled by the **Accumulator Window Duration**. The different control methods for controlling the window duration are shown below. The accumulator output is ready when the Done signal goes high.



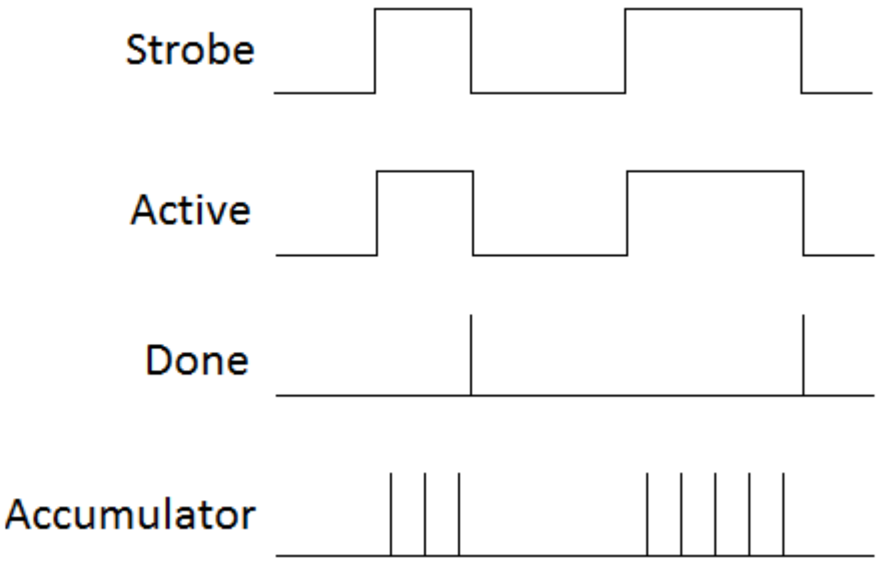
Timer Control Mode

In **Timer** mode, the accumulator runs for a fixed duration when the Strobe input signal goes high.



Counter Control Mode

In **Counter mode**, the accumulator runs for a fixed number of ticks of the accumulator clock. In the above example, the Count is set to 4.



Strobe Input Control Mode

In **Strobe Input** mode, the accumulator start and stop times are determined exclusively by the Strobe gizmo input, which can be variable durations.

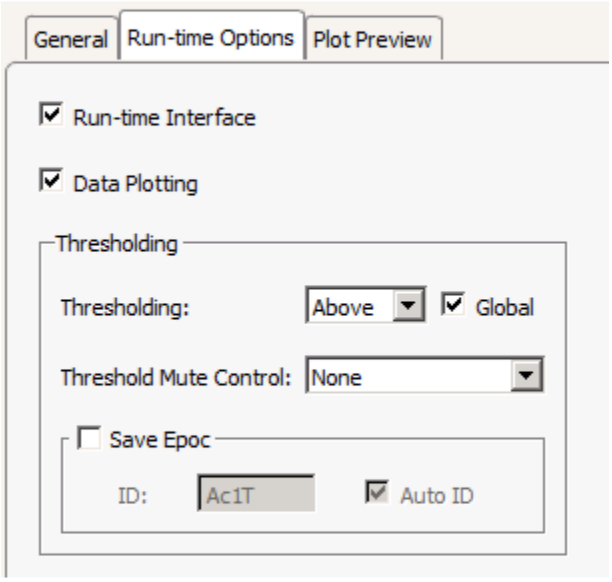
When **Compute Average** is checked, the sum over the accumulation window is divided by the number of accumulation samples to get the signal average.

The gizmo Main and ThrSel outputs latch when the Done signal goes high. When **Dynamic Output** is enabled, the Main and ThrSel outputs also update on each tick of the accumulator clock with the current values in the accumulator.

Run-time Options

The Signal Accumulator has an optional run-time interface that shows a bar graph of the accumulator results. You can optionally set a threshold on this plot to convert this value into a logic signal for decision making. This threshold crossing drives the ThrPass gizmo output.

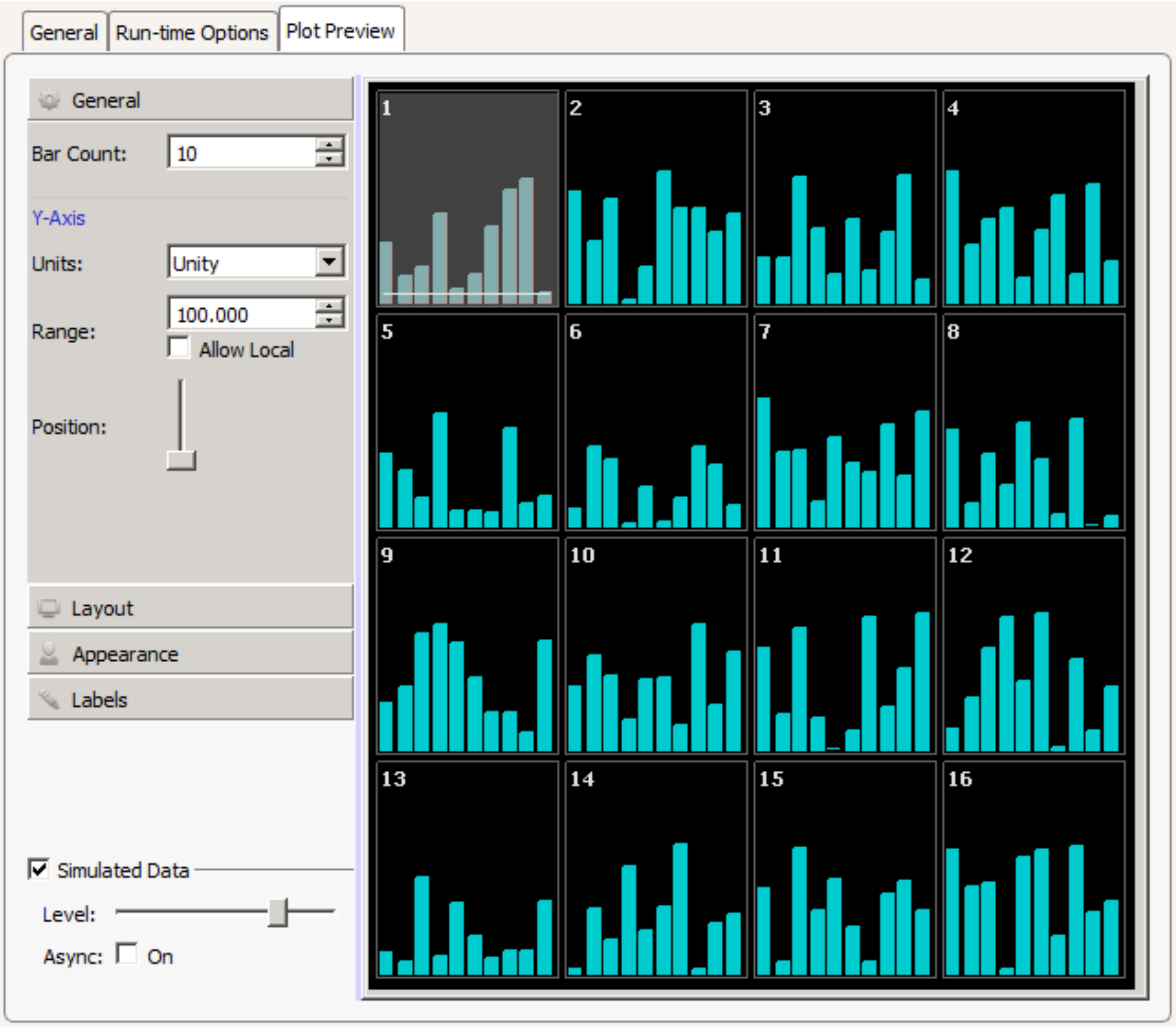
An optional Mute Control allows the user to dynamically disable the ThrPass output at run-time.



Run-time Options Tab

If **Global Thresholding** is selected then any changes to the threshold at run-time apply to all channel thresholds.

Plot Preview Tab



Plot Preview Tab

If **Allow Local** is selected, Ctrl + left-click-drag will change the range of the individual plot and Shift + left-click-drag will adjust the range of all plots.

Note that only the Y-axis range (Shift + left-click-drag or mouse wheel) and offset (Alt + left-click-drag) are adjustable at run-time, as well as the threshold (if enabled). All other visualization settings are adjusted at design- time in the Plot Preview tab.

Important

Accumulator is unique in that the runtime changes modify the experiment setup. Any changes you make to the plot configuration at runtime are available at designtime and vice versa.

Sort Binner

Common Use Cases

Compress sort code output from spike sorting gizmos for fast viewing and further processing. Optionally output to RZ UDP interface for external processing. Use this gizmo to count the number of sort codes that occur on specific channels within a user-specified time window.

Data Stored	
Sort Codes (optional)	Compressed sort code counts as strobe event
Outputs	
Main	Compressed sort code counts as multi-channel integer
StrobeOut	Timing logic pulse when binning occurred

Gizmo Help Slides

Quick Help Slide 1

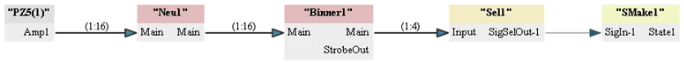
Sort Binner — Takes sort code streams from the spike sorting gizmo and compresses the information into spike counts for further processing

Neural recording with action potentials

Acute neurophysiology

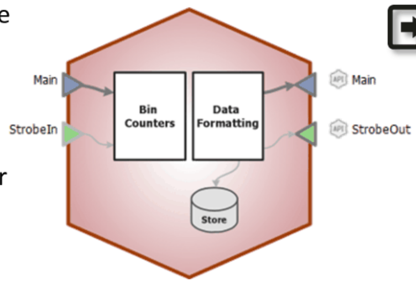
Chronic behavior experiments

64-Channel PCA spike sorting from raw amplifier data stream, sort codes to Sort Binner. Check if # of spikes in bin window is above a certain value with StateMaker.



Main: Input from Spike Sorting gizmo SortCodes output

Strobe: Optional input to control binner timer



Main: Multi-channel integer stream of compressed sort codes

StrobeOut: TTL high when output values are updated

Quick Help Slide 2

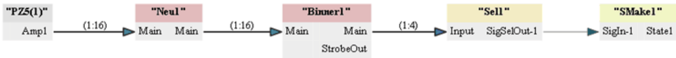
Sort Binner — Takes sort code streams from the spike sorting gizmo and compresses the information into spike counts for further processing

Neural recording with action potentials

Acute neurophysiology

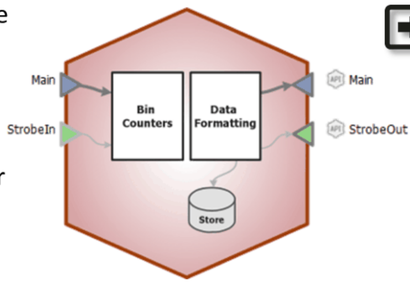
Chronic behavior experiments

64-Channel PCA spike sorting from raw amplifier data stream, sort codes to Sort Binner. Check if # of spikes in bin window is above a certain value with StateMaker.



Main: Input from Spike Sorting gizmo SortCodes output

Strobe: Optional input to control binner timer



Main: Multi-channel integer stream of compressed sort codes

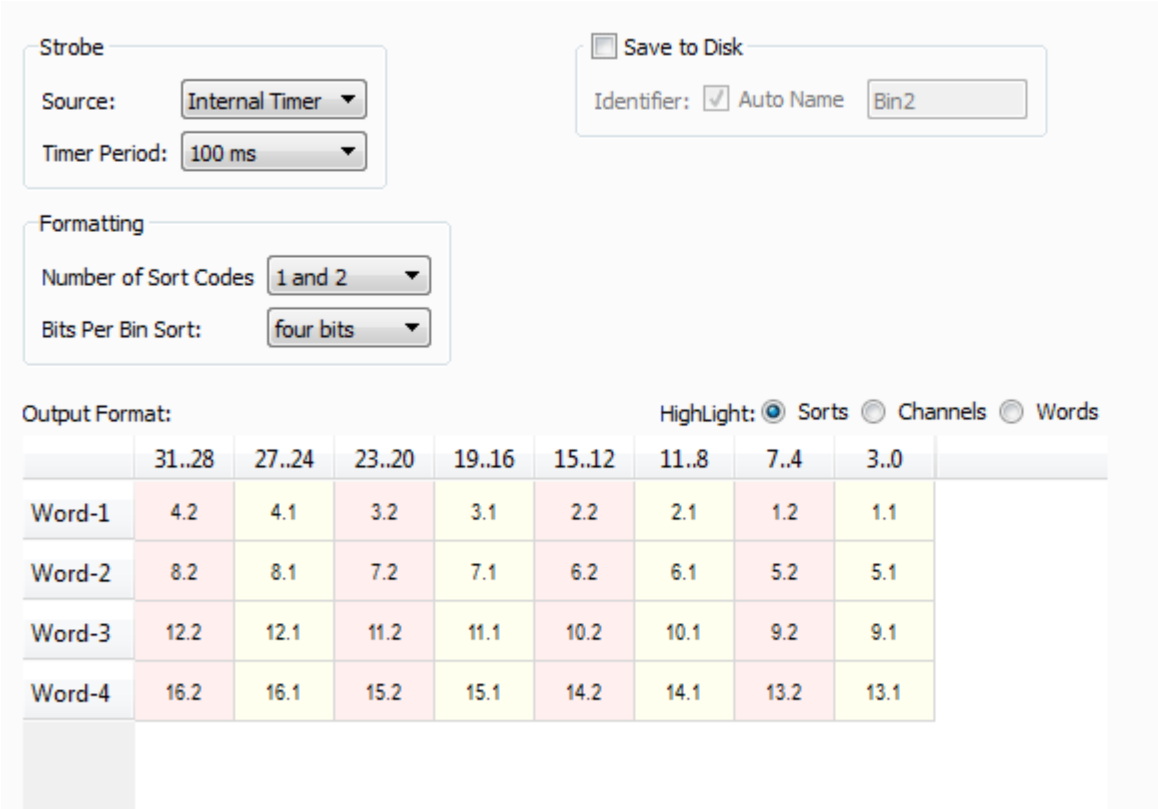
StrobeOut: TTL high when output values are updated

Reference

Sort Binner is designed to work with sort code outputs from spike sorting gizmos. Typically used with the real-time sort code output of a spike sorting gizmo, such as PCA Spike Sorting, Box Spike Sorting or Tetrode Spike Sorting. It's a sort code processor that accepts compressed sort code data and then further compresses the data by counting sort code occurrences within user-defined intervals.

The Main output can be sent to a UDPSend interface or Signal Selector for closed loop control.

Sort Binner Configuration Options



Sort Binner Configuration Options

Strobe

The strobe latches and resets the sort code counter on all channels. It can be a fixed timer (**Internal Timer**) or a logical trigger source from another gizmo or device (**Gizmo Input**).

If using the gizmo input, use the block diagram to choose the **StrobeIn** input after you have committed your selection.

Formatting

Select the number of sort codes to look for on each channel of the incoming data and the number of bits per sort codes you want to use for each counter. Use fewer bits and a shorter strobe period to quickly transfer firing/not-firing information. Use more bits and a longer strobe period to convey a more accurate count of sort codes in between strobes.

Output Format

The table provides a visual reference of how the data is compressed into 32-bit words (integers) and is useful when unpacking the data, for example on the other end of the UDP interface. Words are shown in rows with bits in columns. Each cell contains Channel#.SortCode. Highlight radio buttons are available for fast visual simplification of the format.

In the example above, the gizmo will output a four-channel stream of 32-bit integers (Word 1..4). The first four bits of channel 2 on the Main output will contain an integer count of how many spikes fired on channel 5, that were assigned a sort code value of 1, since the last strobe. Because four bits are used to represent this counter, the maximum count value is fifteen (2^4 -1).

State Maker

Common Use Cases

State Maker is an interface for performing logical tests on single-channel inputs and combining results into output states for storage and controlling/ triggering other gizmos for further signal processing. Use this gizmo when receiving bit codes from external devices, and to make decisions/ process gizmo output values. Used often to trigger store events or strobing other gizmos.

Data Stored	
Epoc (optional)	Selectable: Full, Onset, of Offset Source: Value, Counter, K/M Bits, Inputs
Outputs	
Logic	Multiple, result of logic tests
Integer	Word containing all logic test bits

Gizmo Help Slides

Quick Help Slide 1

State Maker — Perform logical tests on single-channel inputs and combine results into output states for storage and control

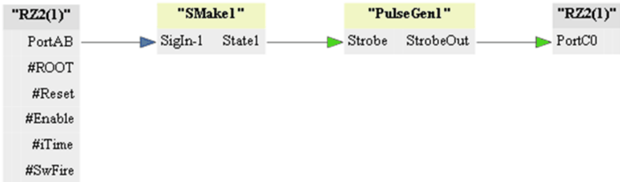
Clean up incoming digital inputs

Monitor lever presses and nose pokes

Behavioral box experiments

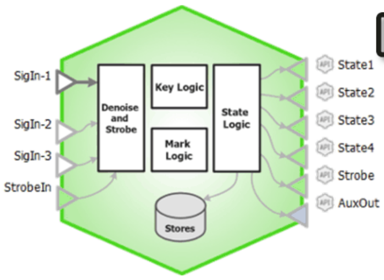
Closed-loop behavior + stim paradigms

Digital inputs on Ports A & B go to State Maker to check which bits are true. If input is specific value, State Maker triggers a pulse train for optogenetic stimulation



SigIn-:* Input signal to perform logic comparisons and tests

StrobeIn: Optional input to control test timing



State:* Output TTL when state is true

Strobe: Output TTL when strobe is true

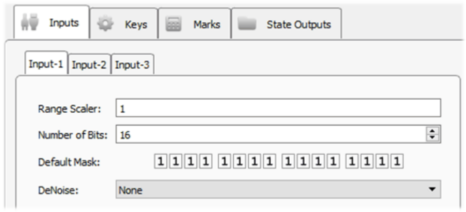
AuxOut: Integer value, can be inputs or integer combination of all keys/marks

Quick Help Slide 2

State Maker

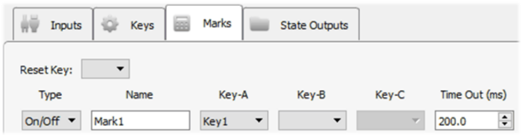
Gizmo Configuration, Data Stored

Configuration of State Maker used in example experiment on previous slide. SM receives a 16 bit word from port AB and reads each bit.

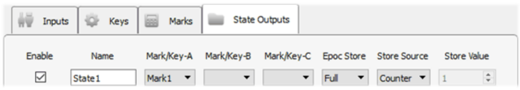


Keys — Truth test comparing input signal to a value (45056)

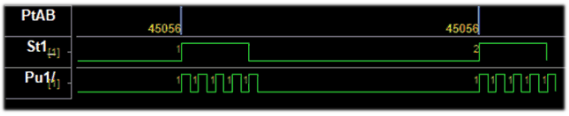
Marks — Combine multiple Keys into a truth test. In this example, when Key1 is true then Mark1 is true for 200ms.



State Outputs — Logic outputs can combine multiple Marks and Keys. In this example, only Mark1 controls State1 output.



*St*_* — Stores onset, offset, or full with optional counter when State is triggered {Epoch}

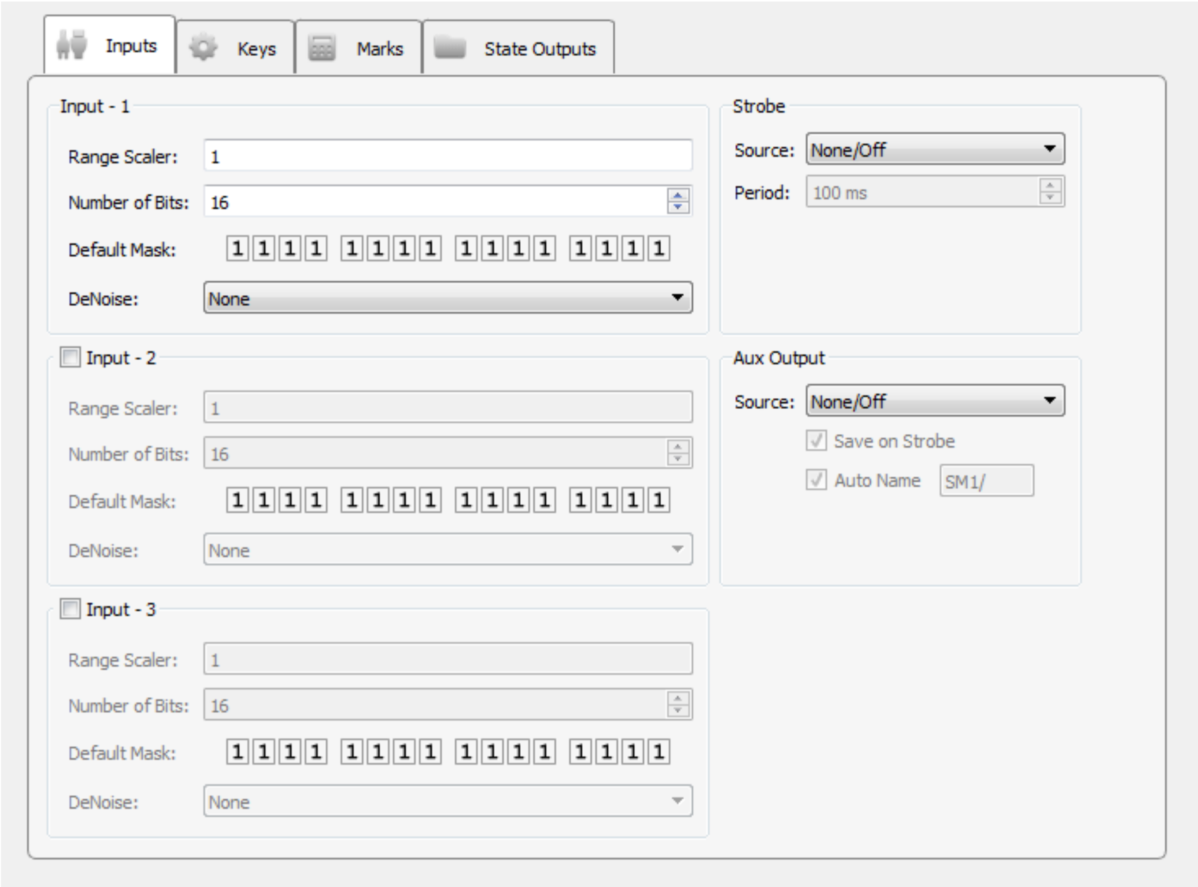


Reference

Inputs are first conditioned to extract the interesting bits. The conditioned inputs are used in logical truth tests, which create 'keys'. Logical combinations of keys are used to create 'marks'. Logical combinations of marks and/or keys are used to create 'states', which can be sent to other gizmos and can be stored to disk. If a Strobe is used, the logic tests are only processed when the strobe is true.

State Maker Configuration Options

Inputs etc Tab



Inputs etc Tab

Input - 1..3

State Maker performs integer-based logic, but can accept any single channel signal (float, integer, logic). Each input is first scaled (**Range Scaler**), which is particularly useful if the input is a floating point signal, such as an eye tracker. Tell it the total **Number of Bits** in the input signal. For a logic input signal, this number will be 1.

Set the **Default Mask** to tell which bits to pay attention to on the input. Set a bit to **X** to ignore it.

DeNoise ensures the value is stable for the specified amount of time before processing it. For example, this is useful if the input is a button press on an external device being read through the digital input on the hardware. This type of signal may 'bounce' when pressed or released, which will create several rapid state changes in that moment, whereas we only want

StateMaker to see a single state change on that input. Adding denoising ensures that the State Maker doesn't process these bounces as individual events and instead waits until the signal is stable for this period of time before changing the input state.

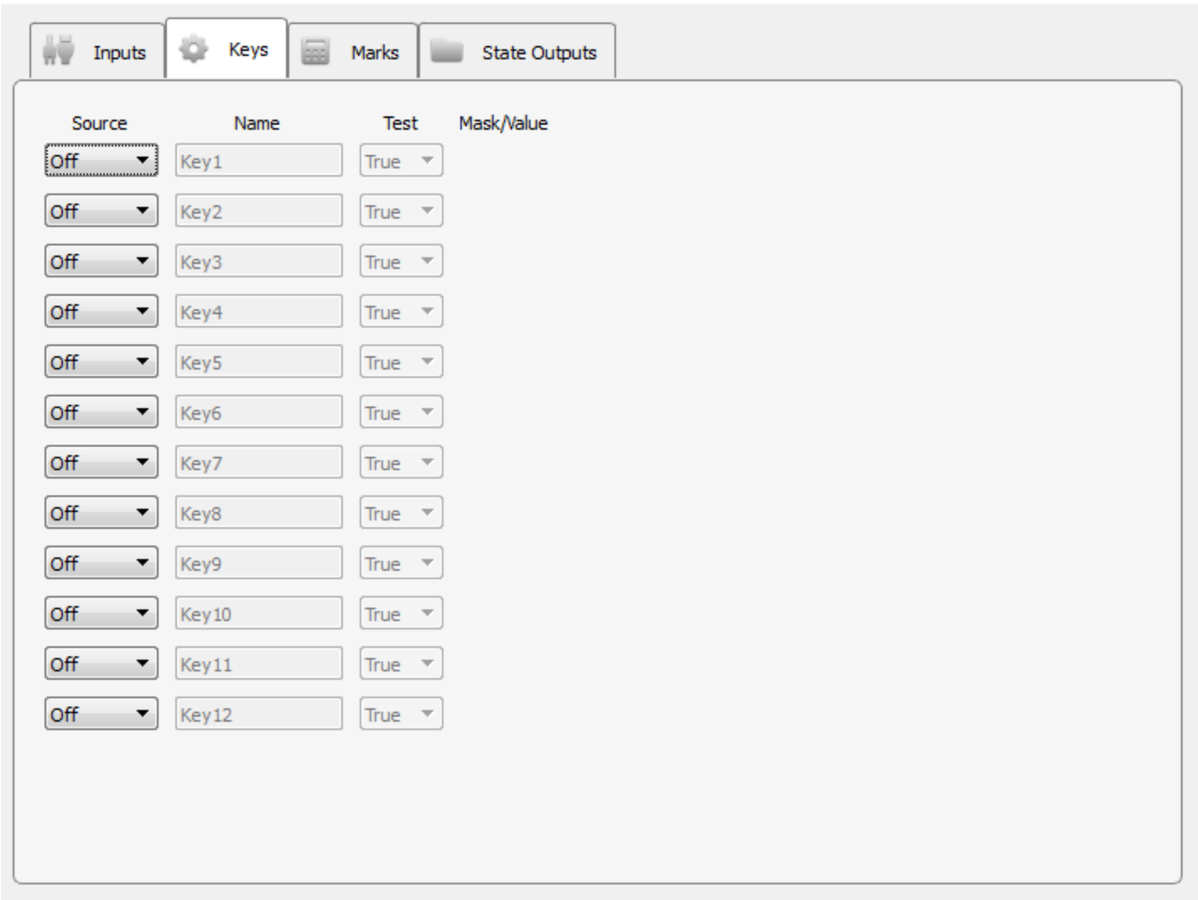
Strobe

By default, State Maker processes the inputs and updates the output states on every tick of the sample clock. Set a **Strobe Source** if you want to control when the logic testing takes place. The strobe can come from an Internal Timer running on the hardware, fed from an additional gizmo input (called **StrobeIn**), or triggered when the value of Input-1 changes (**On Data Change**).

Aux Output

State Maker can output any or all states generated. Set the **Aux Output** to output any of the input lines as a pass through or to output a key, mark, or state output (**Key/Mark/State Word**). If a strobe is used, the **Aux Output** value can also be saved on the strobe.

Keys Tab



Keys Tab

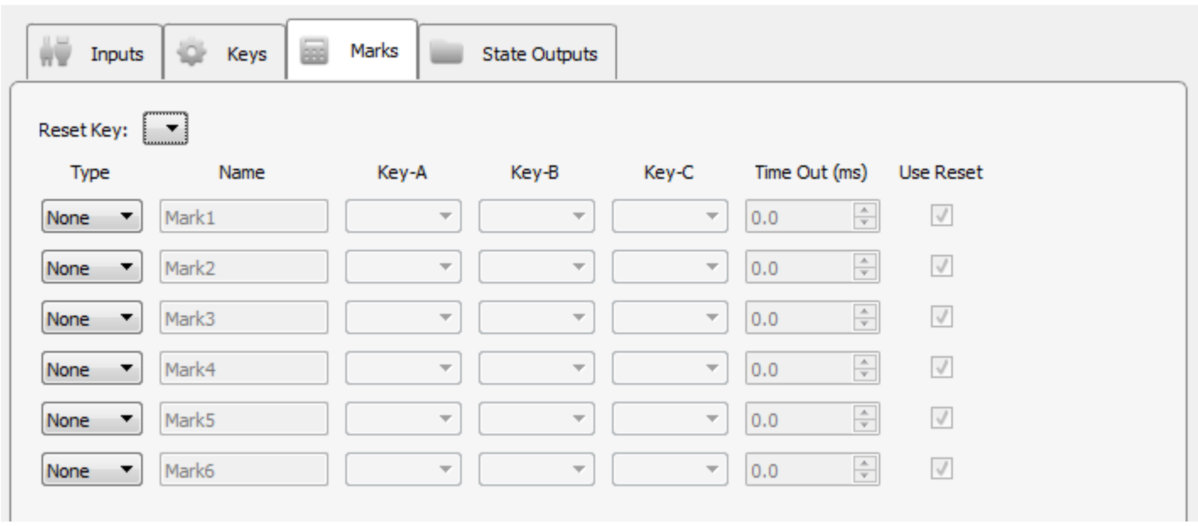
Perform up to twelve logic tests to create 'keys'. Each key uses one of the inputs as its source, and is given a meaningful name that is referenced later on the Marks and State Outputs tabs. It performs a conditional test comparing the key source to the mask/value based on the Test selection.

The Mask tells you which bits to look at or ignore. If a bit in the mask contains an 'x' icon, this bit is ignored during the logic test. If it contains a 0 or 1, that is used for the value test. When Test is 'True', the mask/value is ignored and any source value greater than 0 is considered true.

Click the bit icon to toggle between possible values, that is, 0, 1, or X.

If you would rather enter a number mask/value, right-click on a bit icon in the Mask/Value column to change the data format from binary to decimal or hexadecimal.

Marks Tab



Marks Tab

Use logical combinations of keys to create up to six marks. Each mark uses at least one key as an input and is given a meaningful name that is referenced later on the State Outputs tab.

Keys defined in the Keys tab are listed in the Key-A, Key-B and Key-C drop- downs. At the bottom of the list are the same keys with a '~' prefix - these are the inverse keys, so ~Key1 means 'not Key1'.

Mark Types

The different mark types are described below. Note that any key input (Key-A, Key-B or Key-C) that is left blank is ignored.

On/Off

When Key-A is true, this mark is true and stays true until either Key-B is true or until the **Time Out (ms)** period has expired (if **Time Out (ms)** is non- zero). If **Use Reset** is selected, the Reset Key can also be used to turn off this mark.

And

When all keys specified in Key-A, Key-B, and Key-C are true, this mark is true.

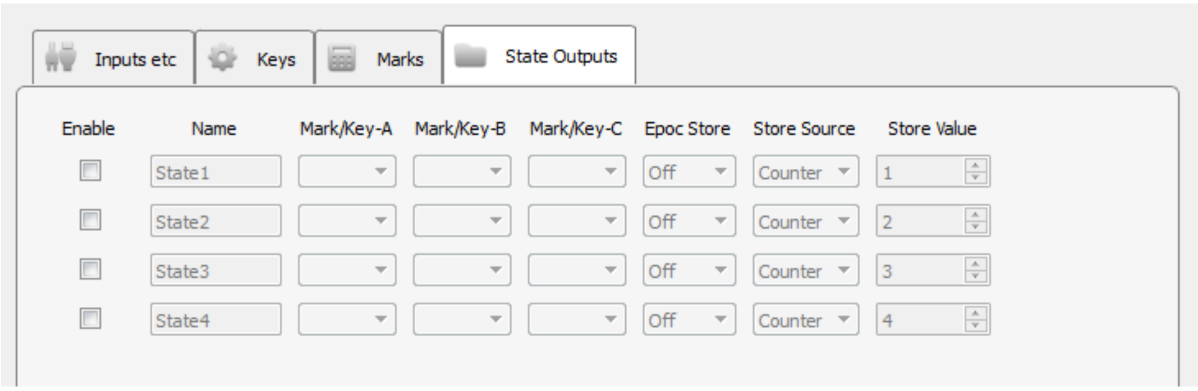
Or

When any key specified in Key-A, Key-B, and Key-C is true, this mark is true.

Xor

When one and only one key specified in Key-A, Key-B, and Key-C is true, this mark is true.

State Outputs Tab



State Output Tab

Use logical combinations of keys and/or marks to create up to four states. Each state uses at least one key/mark as an input and is given a meaningful name that is used when linking to other gizmos and/or storing state values to disk. The states are determined by an 'AND' operation on Mark/Key-A, Mark/Key-B, and Mark/Key-C. Any Mark/Key drop-down left blank is ignored.

You can optionally choose to store the state onset, offset, or onset and offset timestamps into the data tank.

Stream Data Storage

Common Use Cases

Timestamp and store single or multi-channel data when triggered. Use this gizmo to capture behavioral inputs or stimulus parameters to filter and align neurophysiological data.

Data Stored	
Stream	Raw or plot decimated continuous waveforms

Gizmo Help Slides

Quick Help Slide 1

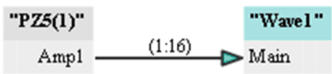
Stream Data Storage — A general-purpose gizmo used to store single or multi-channel data streams. Includes data formatting and scaling options

Record raw data

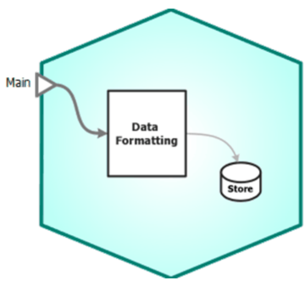
Monitor ADC inputs from extra sensor modalities

Monitor output from gizmos for debugging experimental paradigms

Route raw data signals from your neural amplifier directly to the Stream Data Storage gizmo.



Main: Single or multi-channel input to store



None

Quick Help Slide 2

Stream Data Storage

Gizmo Configuration, Data Stored

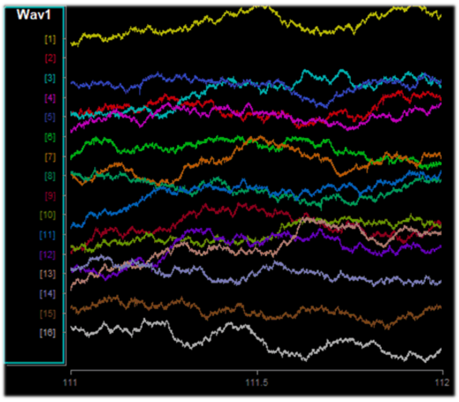
Setup single or multi-channel data streams. Optional control over data format, sampling rate, and value scaling.

Data Format — Use 16-bit integer for reduced data rate. Use PlotDec for plotting only.

Save to Disk — In View Only mode, data is shown on Flow Plot but not saved to disk

Discrete Files — Each channel saves to its own file (SEV). Useful for fast offline processing or pruning bad data.

Wav1 — The continuous streaming data {Stream}



Reference

The Stream Data Storage gizmo is a general purpose data streaming tool that includes options for data format and scaling.

This gizmo stores streamed data in the data tank. Raw waveforms can be saved as 16- or 32-bit floating point values or as integers. The plot decimated waveforms format (PlotDec-16) is a 16-bit representation of the waveform using maximum and minimum values and is used to visualize spike activity.

> **Important**
>
> **PlotDec-16** format is for visualization only. It is not recommended that you store data in this format. It is highly compressed for plotting purposes and has no use in offline analysis.

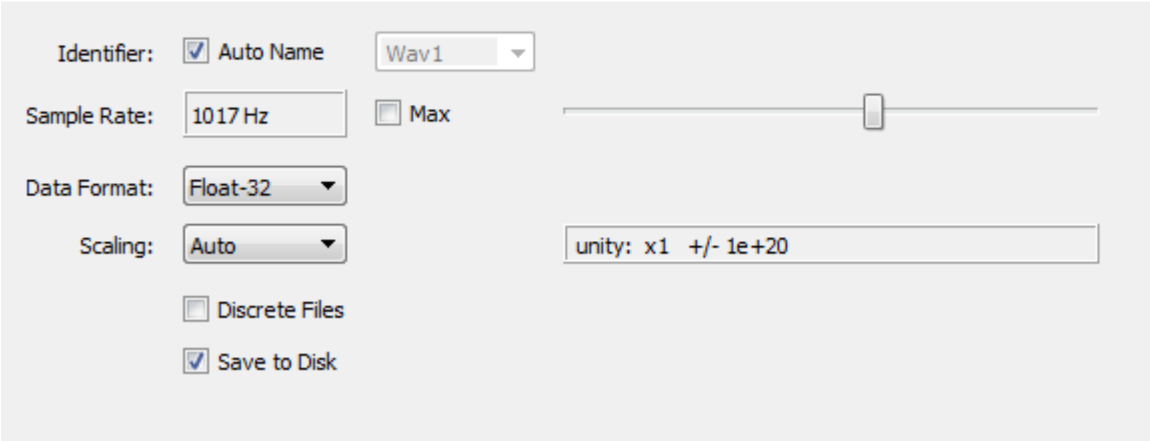
The Runtime Interface

Runtime Plot

A multichannel streamed plot is included in the runtime plot. There are no runtime controls for this gizmo, and it has no outputs available to other gizmos.

See Flow Plot for more information on using and customizing the plot.

Stream Storage Configuration Options



Storage Options

The **Identifier** is used to name the data store that is saved in the tank. It must be four characters in length.

Choose a specific **Sample Rate** for the data store, or set it to **Max** and it will run at the **Master Device Rate** set in the parent RZ or RX processor.

Choose the desired units to apply an appropriate scaling factor to the data.

Select **Discrete Files** to save each channel of data as an individual file (*.sev format) in the data tank. This is useful if you need to clear space by deleting individual channels, or if your post-processing works on individual channels it can be faster to read them this way.

Clear the **Save to Disk** check box to view data in the runtime plots without storing data to the Tank.

Strobed Data Storage

Common Use Cases

Store single values or short segments of data (including pre-trigger data). Includes heat maps and bar plots. Use this gizmo to store streaming data asynchronously or store values/ segments of data around events of interest.

Data Stored	
Scalar	Triggered values of floating point data
Snippets	Triggered snippets of floating point data

Gizmo Help Slides

Quick Help Slide 1

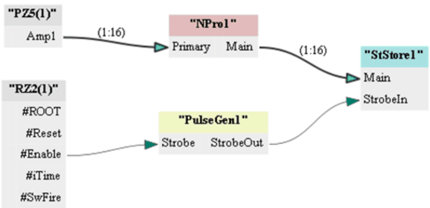
Strobed Data Storage — Store single values or short segments of data (including pre-trigger data). Includes heat maps and bar plots.

Peri-event stimulus response

Asynchronous data storage

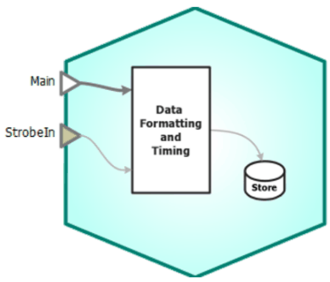
Reduce data block size during long sessions by only recording around events of interest

Route a single channel of filtered data to the StrobeStore. Save a set amount of data whenever triggered by PulseGen.



Main: Single or multi-channel input to store

StrobeIn: Logical input to trigger data saving



None

Quick Help Slide 2

Strobed Data Storage

Gizmo Configuration, Data Stored

Setup flow plots to be snippet, bar graph, or heat map plots. Options for strobing include single scalar, strobe controlled, fixed duration, or continuous.

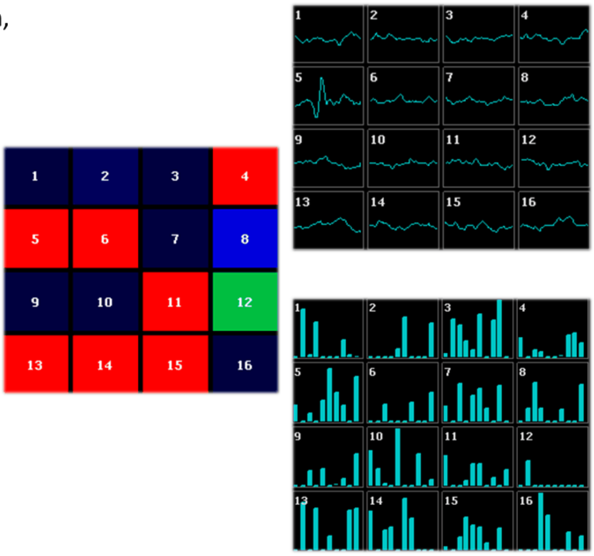
Single Scalar — Save a single scalar value from the data stream when strobed

Strobe Controlled — Save the data stream between the onset and offset of the strobe signal (includes precapture)

Fixed Duration — Save the data for a fixed duration when triggered by the strobe signal (includes precapture)

Continuous — Save the data continuously. Used for very low rate data saving only, otherwise use Stream Data Storage

StS1 — Triggered scalar or snippet {Scalar or Snippets}



Reference

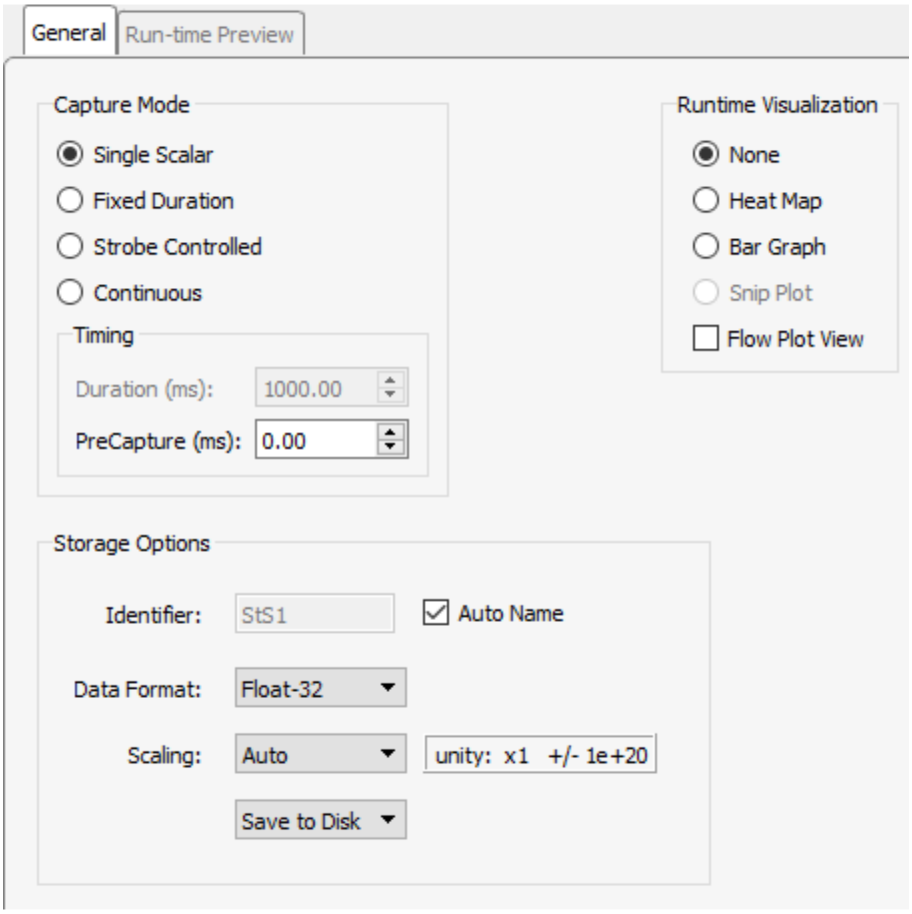
The Strobed Data Storage gizmo stores timestamps and associated event values when triggered. This can be a single value or a short segment of values stored at a specific sampling rate. This gizmo includes advanced runtime visualizations.

The Runtime Interface

A strobe plot can be added to the Flow Plot for visualization. See Flow Plot for more information on using and customizing the plot.

Additional visualizations are available depending on the Capture Mode. See Runtime Visualization for more information on using and customizing these plots.

Strobed Storage Configuration Options

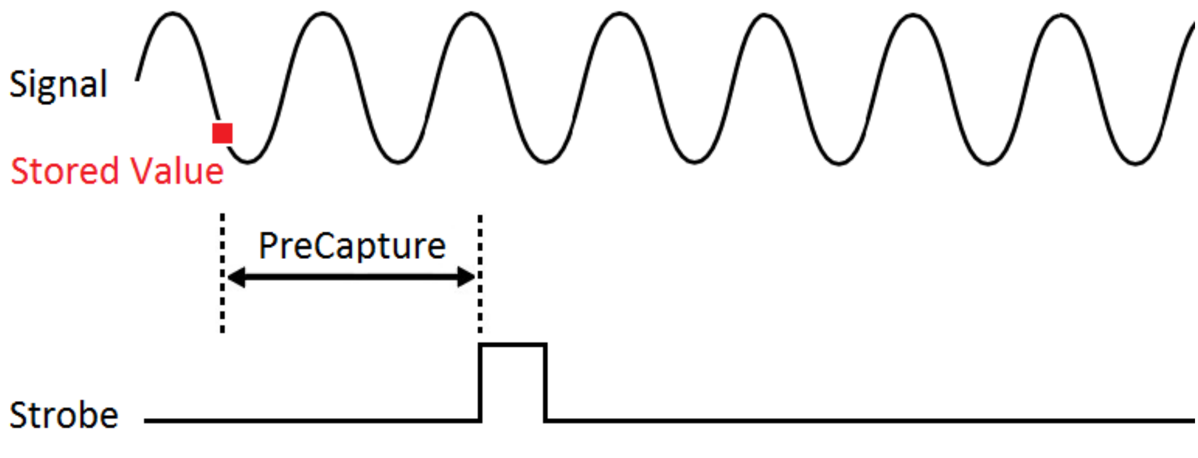


Strobed Storage Configuration Options

Capture Mode

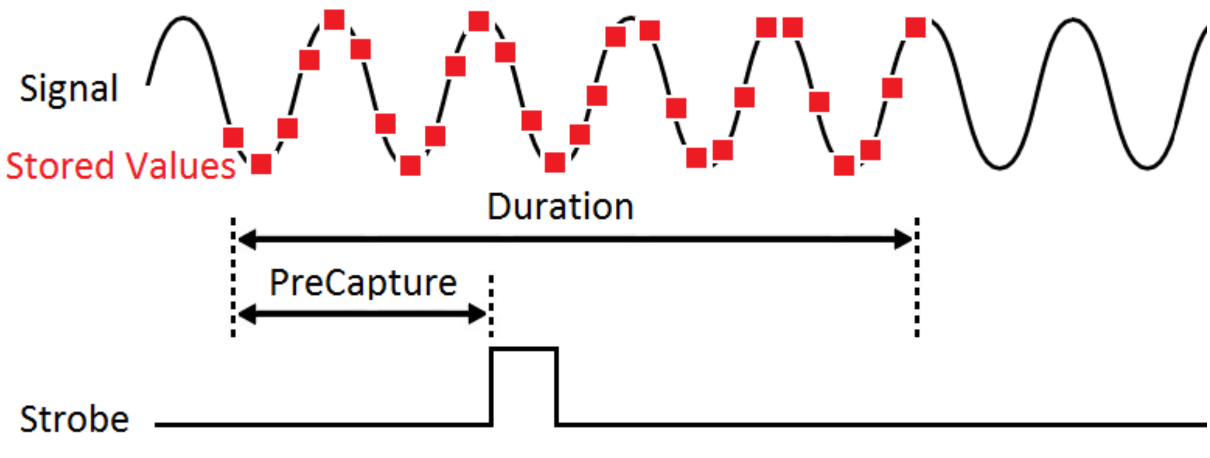
The Timing and Storage Options have different meaning depending on which capture mode is selected.

1. **Single Scalar** stores a single data point on the rising edge of the Strobe input. If you want to record the value at some fixed time before the trigger occurs, set **PreCapture (ms)**. The timestamp stored in the data tank will also be delayed by this amount.



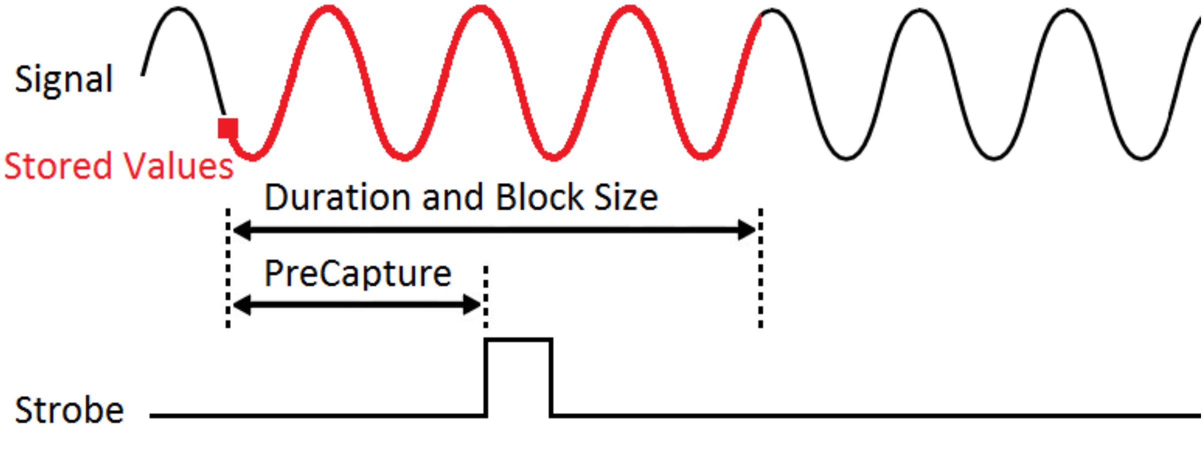
Single Scalar Timing Diagram

2. **Fixed Duration** stores a fixed number of points on the rising edge of the Strobe input. Set the desired sampling rate of the acquired data. The format of the data stored into the tank is optimized automatically for you depending on the sampling rate.
    a. If the rate is below 10 Hz, each stored data point will have a timestamp and value (it will be stored as timestamped scalar values).



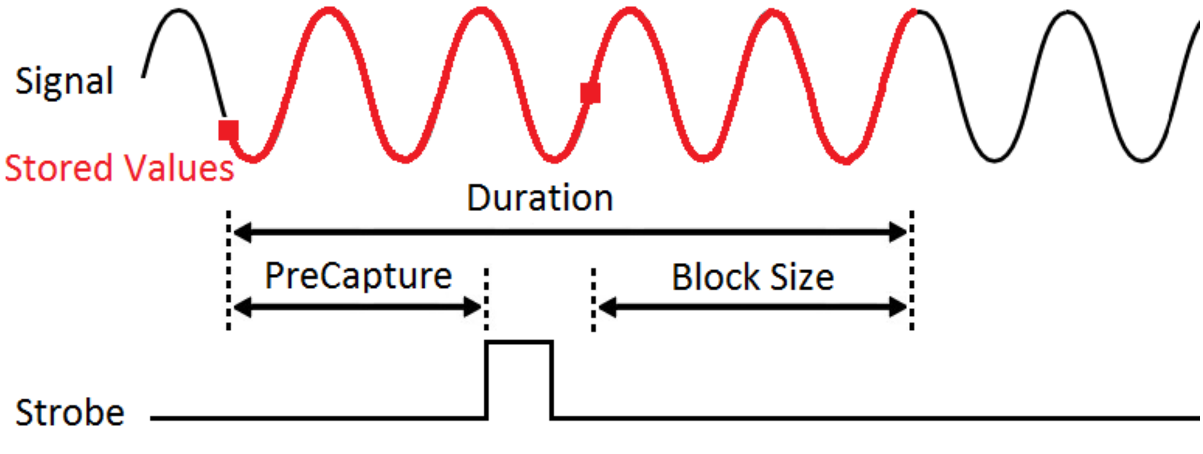
Fixed Duration Scalar Timing Diagram

    b. If the rate is 10 Hz or above, the data is recorded in short blocks (Record Type Block) that include a single timestamp and chunk of points, determined by the **Block Size** setting.



Fixed Duration Block Timing Diagram

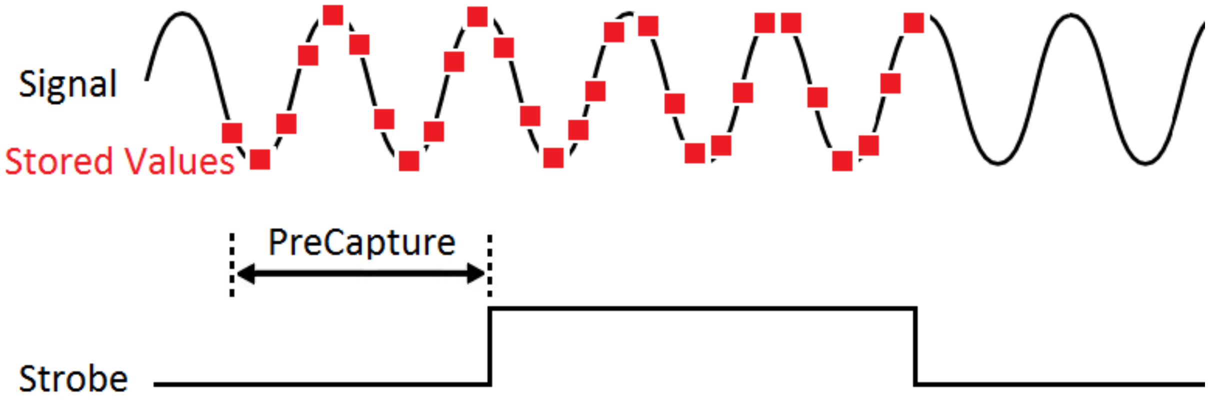
If the duration is longer than the optimized maximum block size, the recordings are broken up into smaller blocks, determined by the **Blocks per Capture** setting. The diagram below shows two sub-blocks.



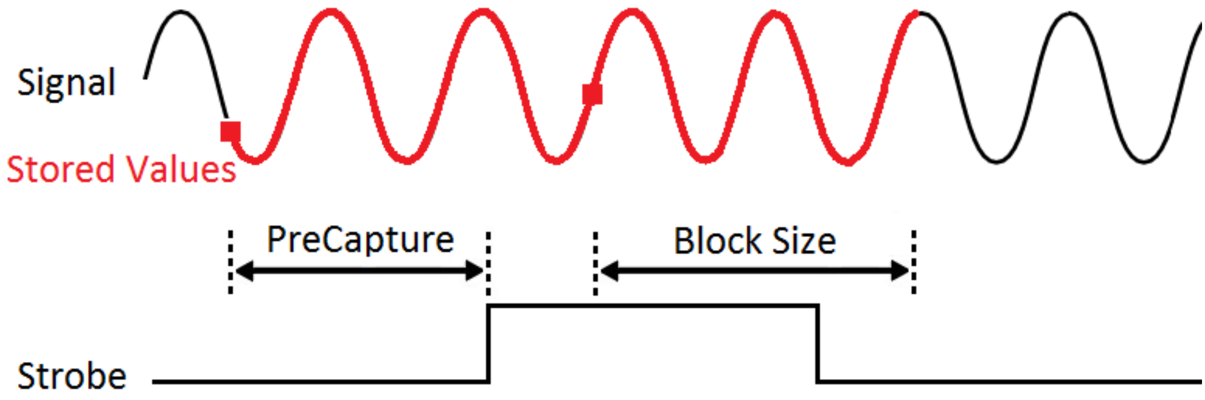
Fixed Duration Sub-Block Timing Diagram

All of the automated settings in the **Storage Details** options are designed to optimize data transfer from the hardware back to the PC. You can override these defaults by checking the **Storage Details** box, though this is not typically recommended.

3. **Strobe Controlled** stores values only when the Strobe input is high. The sampling rate determines the record type (Scalar or Block).



Strobe Controlled Scalar Timing Diagram



Strobe Controlled Block Timing Diagram

When the record type is Block, the last recorded sample will typically be beyond the end of the strobe because the Block Size is always fixed while the Strobe input duration can be variable.

4. **Continuous** mode ignores the Strobe input and data is continuously recorded into the tank at the specified sampling rate.

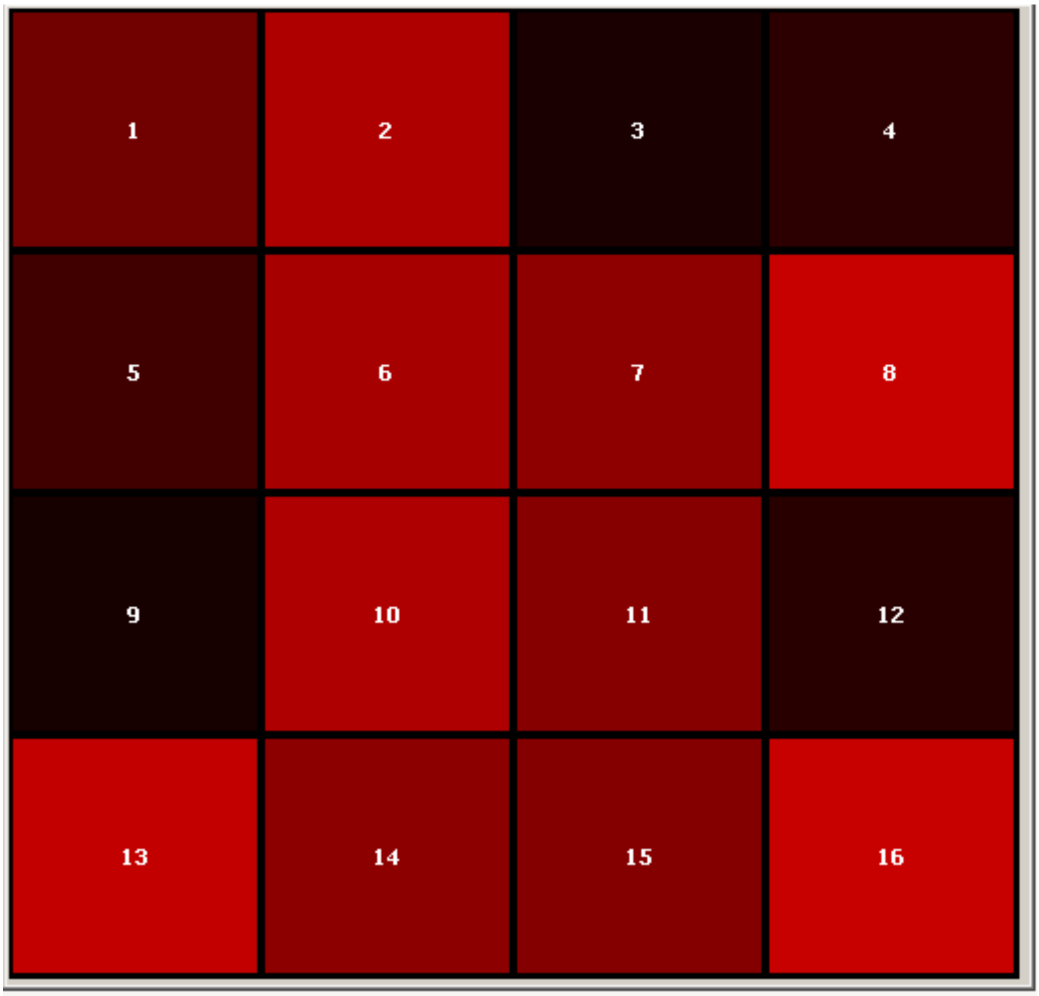
> **Important**
>
> For high sampling rates of continuous data above 30 Hz, the Stream Data Storage gizmo is recommended instead.

Runtime Visualization

Choose what type of plot(s) to see at run time.

1. **Heat Map**

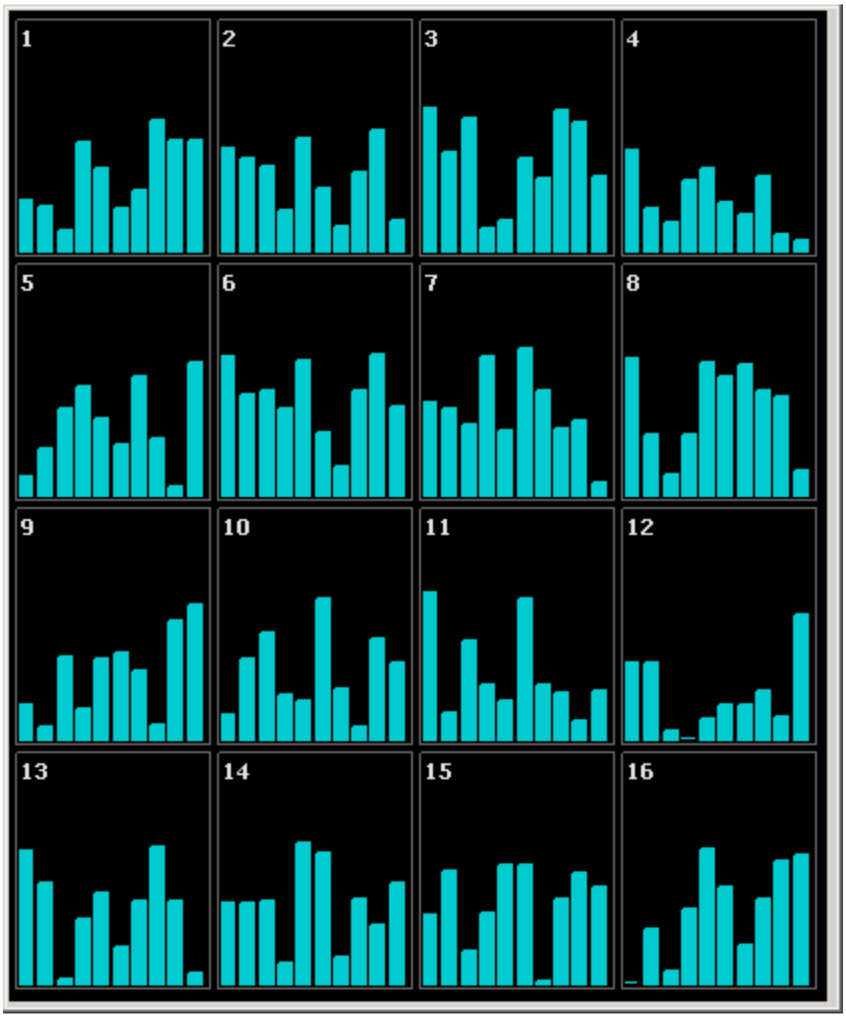
   For Single Scalar values, display a grid with value mapped to color intensity.



Heat Map

2. **Bar Graph**

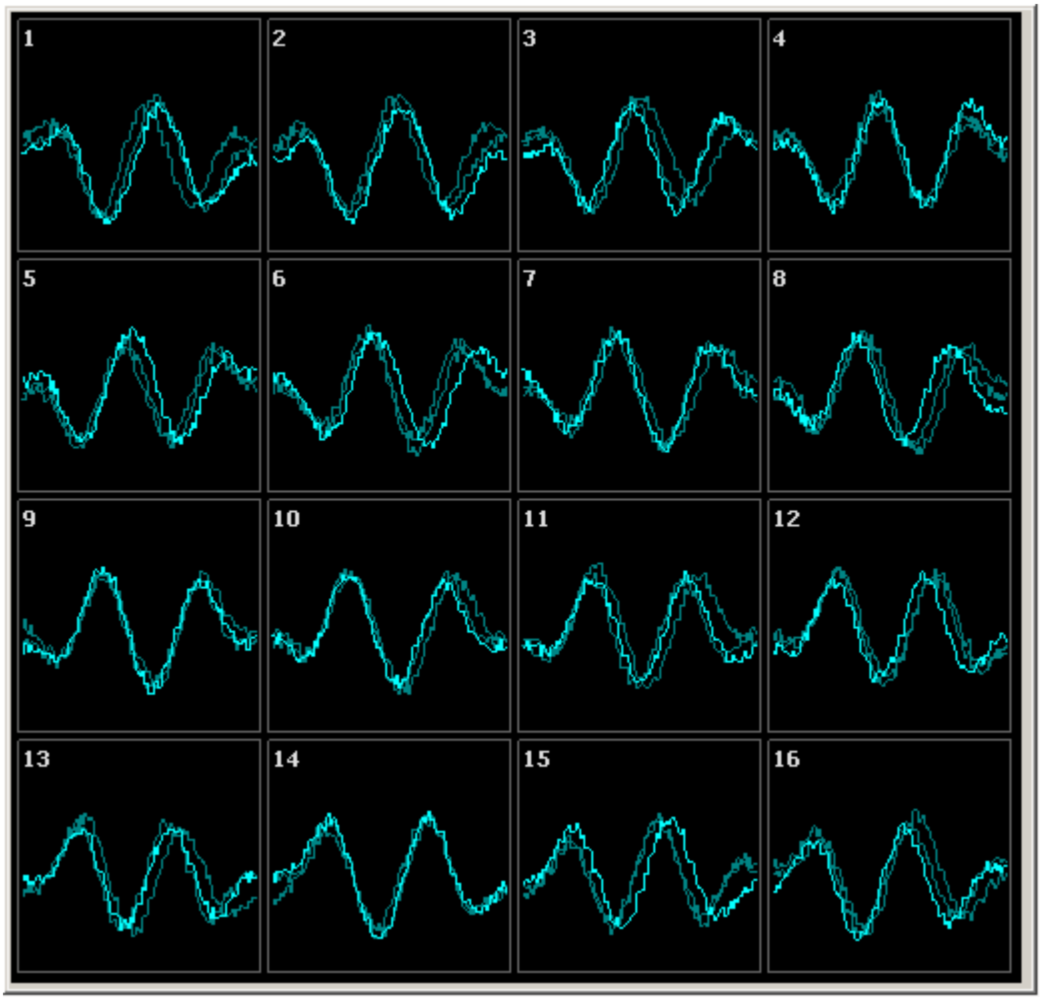
   For Single Scalar values, display a bar graph of the previous N values.



Bar Graph

3. **Snip Plot**

   For Fixed Duration storage, plot the previous N waveforms and highlight the current waveform.



Snip Plot

Flow Plot View

Select this to also show the corresponding data store in the Flow Plot window (recommended).

Tetrode Spike Sorting

Common Use Cases

Real-time filtering, cross-channel tetrode spike detection and classification in a fully customizable 2D feature projection. Use this gizmo for sorting spikes using tetrodes. Commonly used for cell isolation, tetrode sorting provides high spatial localization of nearby units.

Data Stored	
Snippets (optional)	Timestamped spike waveforms
Stream (optional)	Plot decimated waveforms
Outputs	
Main	Filtered, multi-channel floating point signal
Sort Codes	Multi-channel integer signal containing compressed sort codes

Gizmo Help Slides

Quick Help Slide 1

Tetrode Spike Sorting — Apply real-time filtering, cross-channel tetrode spike detection and classification using 2D feature projections

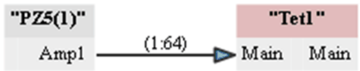
Neural recording with action potentials

Microdrive experiments

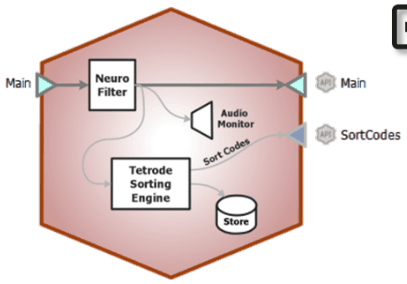
Acute neurophysiology

Chronic behavior experiments

64-Channel Tetrode spike sorting from raw amplifier data stream



Main: Input fed from neural amplifier or channel mapper



Main: Filtered signal output. Typically unused because data stream can be saved directly in gizmo

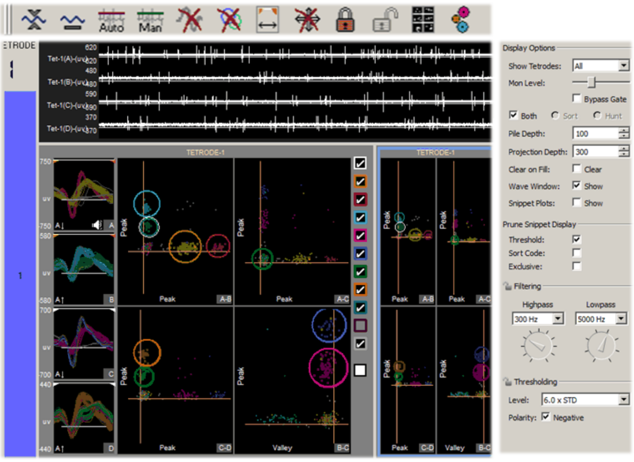
SortCodes: Multi-channel integer values containing sort code information. Typically goes to Selector or Sort Binner for further online processing

Quick Help Slide 2

Tetrode Spike Sorting

User Interface, API, Data Stored

Dynamic control of filters, thresholding, and clustering allows for advanced clustering of thresholded snippets.



FreqHP/FreqLP — Corner frequency control of filters {Read/Write}

ThreshLock — Lock the threshold level (per channel) {Write}

ThreshMode — Switch between manual and dynamic thresholding (per channel) {Read/Write}

Threshold — Change the threshold level (per channel) {Read/Write}

pTe1 — Filtered plot-decimated data stream {Stream}

eTe1 — Captured spike snippets with assigned sort codes (if Hardware Sort enabled) {Snippets}

Reference

The Tetrode Spike Sorting gizmo performs filtering, thresholding and online tetrode feature space spike sorting and storage on multi-channel neural signals at sampling rates up to 50 kHz.

Data Storage

This gizmo generates two types of data for storage: snippet data (includes timestamp, short waveform, and sort code) and plot decimated data streams. The stream data generated by this gizmo is a highly decimated version of the waveforms that keeps local maximum and minimum values of the filtered signals, which makes it ideal for visualizing high frequency spike activity on a computer monitor with a fixed number of pixels.

When a waveform crosses a threshold on any channel in tetrode, a snippet on all four channels in that tetrode is recorded. The four snippets are concatenated and stored in the data tank as one large snippet, with a timestamp and a sort code. The sort code is determined by visual spike sorting in the runtime interface.

In plots and in the data tank, each type of data is designated with a prefix: 'e' for snippets and 'p' for streams. You can opt to save only snippets or to disable storage in the gizmo's configuration settings. The sort codes can be configured as an output to be used in other gizmos.

Threshold Detection

At runtime, candidate spikes are detected based on a calculation of the deviation of a waveform from its RMS. By default, the timestamp and position of the waveform in the snippet is dependent on the time of the threshold crossing for the signal. An alternative setting allows waveform timestamp and positioning to be determined by the waveform's highest peak, aligning snippets to their respective peaks. By default, detection is automated and you can make adjustments in the threshold control plot in the runtime window.

Spike Sorting

Each channel within a tetrode is displayed in a separate snippet waveform subplot. Events are projected onto a 2D space by first calculating user-selected metrics for one or two channels and then mapping one metric against another. Up to four 2D feature projections can be used to visualize spike clustering. Users may select from the following metrics: peak, valley, height, energy, non-linear energy, average, area and Slope. User-defined circles in each projection plot determine each cluster's boundaries. Snippets falling inside a circle are given a sort code corresponding to that circle's color.

The interface works in two modes:

Hunt Mode

In hunt mode the projection plots default to peak vs. peak for all six combinations of tetrode channels to provide a general overall picture of activity. Use this mode during electrode placement to search for active neurons.

Sort Mode

After the electrode has been placed, use sort mode to choose new metrics for the projection plots and add sort circles to these plots.

This gizmo allows simultaneous recordings from multiple tetrodes. The multi- channel input stream must be arranged in groups of four; each group corresponding to one physical tetrode (a Mapper gizmo may be used, see Mapper).

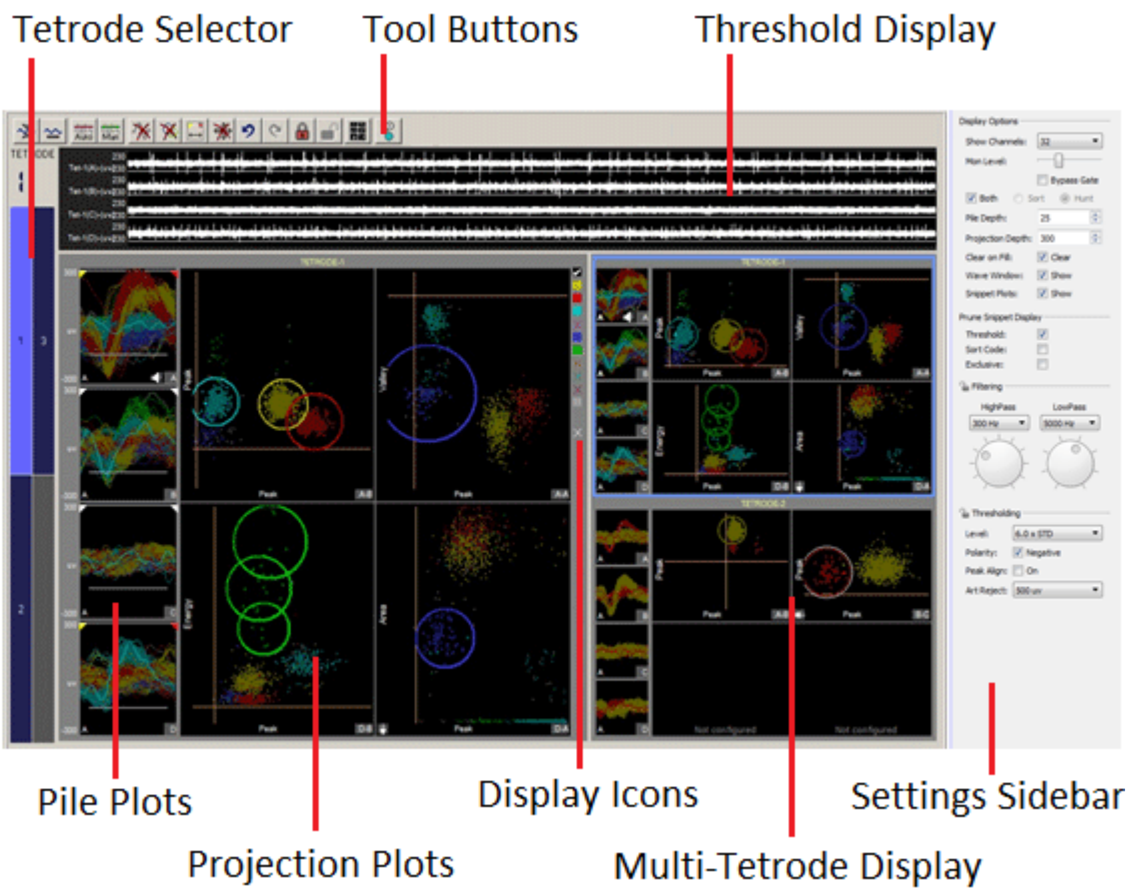
Settings for configuring the maximum number of sorting circles per projection, thresholding method and window width of the snippets can be found on the Sorting Tab in the Options area of the designtime interface.

The Runtime Interface

Runtime Plot

Streamed waveform and Snippet plots are added to the runtime window for visualization.

Tetrode Spike Sorting Window



The runtime window includes:

Display	Description
Tool Buttons	Perform actions that are global to all channels.
Threshold Display	Displays the plot decimated waveforms for the currently selected tetrode and the threshold marker for each channel. When automatic threshold tracking is active the threshold bar is locked.
Tetrode Selector	Selects the active tetrode and indicates channel status. Gray indicates the channel is locked and you can't change the sorting parameters.
Pile Plot	Displays candidate spikes for the active tetrode. Indicators on the bottom left corner denote scaling and threshold tracking mode ('A' for automatic, 'M' for manual). The letter on the bottom right indicates the channel's position within the tetrode (A, B, C, D), which is used in the Projection Plots. A speaker icon indicates the channel that is currently playing out of the audio monitor. Hold down the **C** key and click a pile plot to send that channel to the audio monitor.
Projection Plots	Displays the active tetrode in several projection plots for easy comparison of candidate waveforms and visual sorting. Double-click one of the plots to choose the features/ channels used for its projection. Hold down the **Ctrl** key and click-and-drag to draw a sorting circle on the plot. Hold down the **Alt** key and click-and-drag to draw an arbitrary shape that will be converted into sorting circles. If the origin point is not in view, and arrow in the bottom left corner indicates the direction to the origin.
Display Icons	Choose which sort codes are displayed in the pile and projection plots. The bottom icon turns off any custom highlighting.
Multi-Tetrode Display	Displays each tetrode in a smaller version of the projection plots to allow the user to monitor all tetrodes while working with the active tetrode.
Settings Sidebar	Includes settings for display options, filtering, and threshold settings.

Zoom and Pan

You can zoom any plot to see more or less detail without affecting the actual data.

To change the zoom level, hold down the **Shift** key and click-and-drag the pointer up or down.

To reset the zoom level, hold down the **Shift** key and double-click the pointer within the display area.

To pan in snippet plots, hold down the **Alt** key and click-and-drag to move the snippets vertically.

To pan in projection plots, click-and-drag the pointer to move the view.

Display Scale

To make it easier to see waveform shapes for channels with lower magnitude, you may scale individual channels manually or normalize all channels to fit to a similar scale, all without altering the data being stored.

To normalize all channels, click the **Auto Scale** button in the toolbar and choose to normalize the display. Each channel is scaled individually to fit around 80% of the signal's vertical size in each plot. An up or down arrow is displayed in the bottom left corner of the plot or subplot to indicate whether the display has been scaled up or down. This does not change the scale of the feature space.

To adjust the scale of a single channel, press and hold down the **Ctrl** key, and click-and-drag the pointer up or down in the pile plot. While adjusting the display scale, the numeric value in the lower right corner of the channel plot indicates the new scale value.

The gizmo stores two sets of scale factors, one set for sort mode and another for hunt mode. Each set (sort or hunt) of scaling information includes a scale factor for the tetrode and any individual scale factors set for individual plots. This allows you to switch between modes without rescaling or losing scaling information.

To reset the scale for all channels, click the **Reset Base Scale** button. This does not remove any zoom applied to a plot.

To return all channels of a single tetrode to their base scale, right-click in the wave window and select **Reset Scaling - Tetrode** from the menu.

To return a single channel to its base scale, right-click the desired pile plot and select **Reset Scaling** from the menu.

Scaling the Projection Plots

In addition to being scaled when all plots/channels are scaled, the projection plots can be scaled for each tetrode or as individual plots on the shortcut (right-click) menus.

The projection plots also have a base scale which is computed as a reasonable estimate based on the metric combinations and typical data sets.

Because the 2D clusters don't always fit into nice circles for sorting, the projection plot axes can be independently scaled in order to skew the visual data set so that it does fit into a circular boundary.

To independently scale each axis, hold down **Ctrl + Alt** and click-and-drag the pointer to the left or right to scale the x-axis of the project plot, or drag up or down to scale the y-axis.

To reset the independent scaling for all projection plots in the current tetrode or all tetrodes, click the **Reset 2D Plot Independent Scaling** button.

Highlighting Traces

By default, the most recent trace acquired is highlighted in all plots throughout the tetrode display area. Alternatively, a group of traces that are of interest can be highlighted.

To highlight a group of traces, hold down the **A** key and drag the pointer across the desired traces in any pile plot. The selected pile plot traces and their corresponding dots in the projection plots will be highlighted.

This can be repeated to add more selected traces. To remove a group of traces from the highlighted selection, repeat this procedure with the **S** key.

To clear all custom highlighting from the pile and projection plots of the active tetrode, click the bottom display icon (to the right of the projection plots).

Settings Sidebar

Display Options	Description
Show Channels	Select the number of channels to display in the multi-tetrode display.
Mon Level	Slide the indicator to adjust the level of the audio monitor output, when enabled.
Bypass Gate	A noise gate on the audio monitor removes background noises so only the spikes are heard. Select this check box to turn off the noise gate.
Both, Sort, or Hunt	Choose to apply the settings below in sort, hunt or both modes.
Pile Depth	Enter a number to set the maximum number of events displayed in pile plots. The oldest waveform traces are removed as new events are added.
Projection Depth	Enter a number to set the maximum number of events displayed in projection plots. The oldest waveform traces are removed as new events are added.
Clear on fill	Select the check box to refresh pile plots, clearing all traces for a given channel whenever the pile depth is reached on that channel. The same applies to projection plots when the projection depth is reached.
Wave Window Show	Select to show the wave window.
Snippet Plots Show	Select or show pile plots in the multi-tetrode display.

Prune Snippet Display	Description
Threshold	Select to show only snippets that crossed the threshold for the channel on which they occurred.
Sort Code	Select to show only snippets from channels that are used as metrics for the projection plots that have sort circles in them.
Exclusive	Select to show traces only in the pile plot in which the corresponding sort code first appears. Note: this will always be the pile plot for the first channel of the tetrode, unless used in conjunction with the threshold or sort code pruning options.

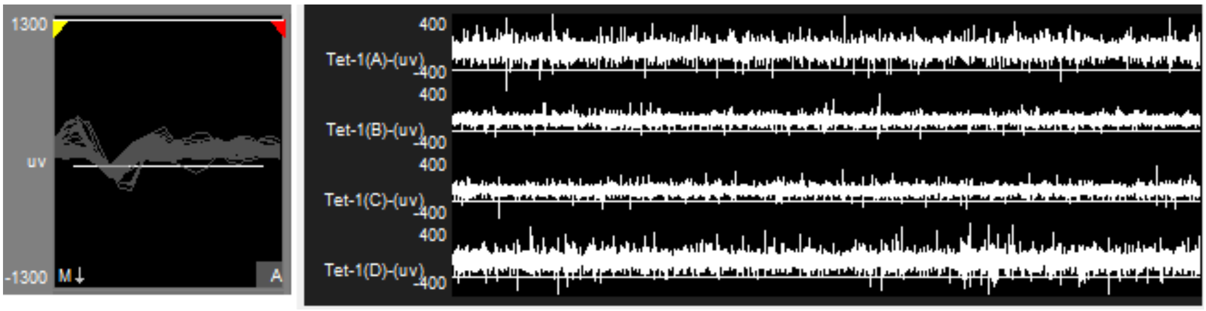
Filtering Options	Description
HighPass/LowPass	Set the highpass and lowpass digital filter settings. The filter is applied to the data before threhsolding, sorting, or visualization

Thresholding Options	Description
Level	Set the automatic threshold level for spike detection, in number of standard deviations from the baseline (smoothed over a 3 second window)
Polarity	Set automatic threshold search polarity, either positive or negative
Art Reject	When artifact rejection is enabled in the configuration options, sets the artifact rejection level in microvolts. If any sample of the candidate waveform is above this level, the waveform is ignored

Threshold Control

Click the **Auto Threshold** button to initiate automatic threshold tracking on all unlocked channels. If auto thresholding is enabled in the designtime interface, real-time tracking will begin on all channels, otherwise the channels will remain in manual threshold mode and the threshold will be set based on a one-time calculation using the current window data and the **Thresholding Level** and **Polarity** settings.

Click the **Manual Threshold** button to enable manual thresholding on all unlocked channels. In manual threshold mode, the threshold bar can be adjusted by clicking and dragging the white bar in the threshold display or pile plot.



Pile Plot (left) and Wave Window (right), Manual Threshold Mode

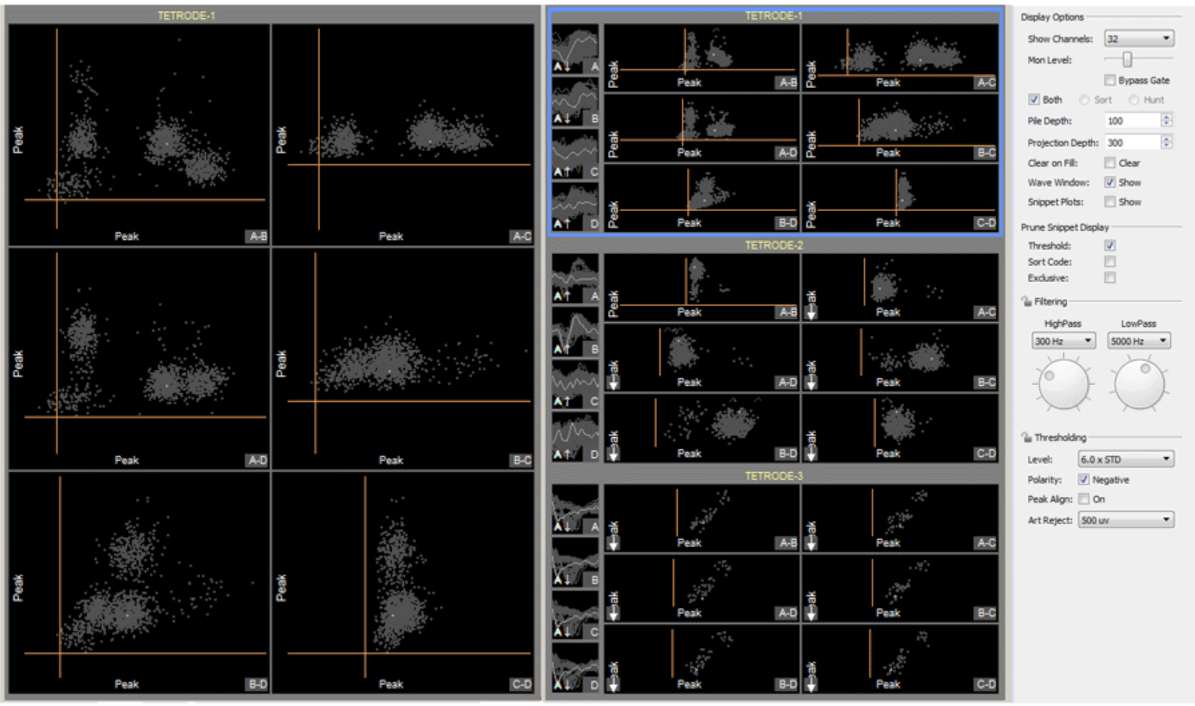
You can also right-click the pile plot at the desired threshold location and choose **Set Threshold Here** from the shortcut menu to move the threshold to that location on one channel. You have the option to apply this new location to all channels in manual thresholding mode.

Right-click the pile plot or threshold display and use the **Auto/Manual Threshold** options to change the threshold mode of an individual channel.

Hunt Mode

Hunt mode is designed for use during electrode placement. In this mode, the feature projections show peak vs peak for all electrode combinations to provide a general overall picture of activity.

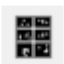
By default, the runtime interface is in sort mode. To turn on hunt mode, click the **Hunt** button. All sorting features are disabled in this mode, but scaling and other features are available.



Hunt Mode

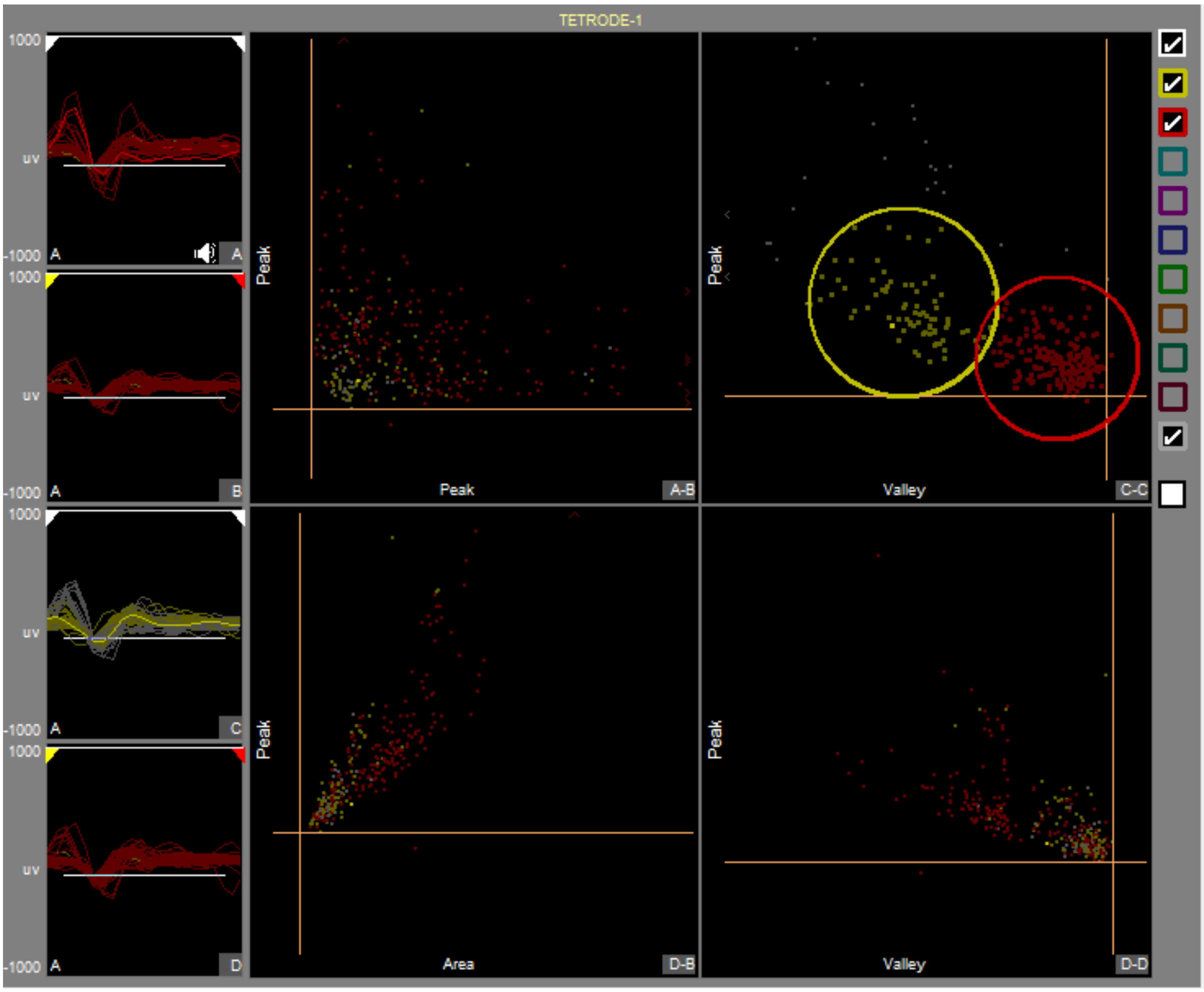
The number of events that are shown in the snippet plots (**Pile Depth**) and projection plots (**Projection Depth**) can be configured in the settings sidebar.

To clear all events from the display, press the **Spacebar**.



You can transfer projections from hunt mode to sort mode to speed up projection configuration. Press the **Ctrl** key and click the desired projection(s). You can select multiple projections across multiple tetrodes. Then click the **Hunt** button to turn hunt mode off. You'll be asked to confirm your selections. The display returns to sort mode with the selected projections.

Sorting in the Active Tetrode Plot Display



Active Tetrode Display

The active tetrode display provides an interactive space for online cluster- cutting. Once reasonable thresholds are set, snippets will appear in each of the four snippet plots. The projection plots are created by applying metrics to the waveforms in these pile plots and then plotting one metric versus another in an X-Y plane.

When you are satisfied with the defined sorts, you can send the sorting parameters to the hardware by clicking the **Hardware Sorts** button and sort codes will be applied to new data as it is acquired in real-time. This toolbar button **must** be 'pressed' for online sorting to take place on the hardware.

You can lock the plots by clicking the **Lock** button to protect them from modification.

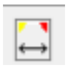
Pile Plots

Restricting Metric Calculations to a Narrow Window

The marker at the top of the snippet plot indicates the window of snippet data that is used for metric calculations for the selected projection plot. This is called the min-max interval and can be adjusted on-the-fly.

This does not change the width of the window in data storage, which is defined in the configuration options.

Click-and-drag the yellow or red indicators to the desired position to adjust the min-max interval.



To reset the min-max interval for one channel, right-click in the pile plot and select **Reset Min-Max Interval**. To reset the min-max interval for more channels, click the **Reset Snippet Plot Max Min Intervals** button.

Projection Plots

The main area of the active tetrode display is divided into four 2D projection plots for dynamic visual spike sorting. Each snippet appears as a single dot in 2D space of one metric plotted versus another. By default, the top left plot will display the peak of the first channel against the peak of the second channel.

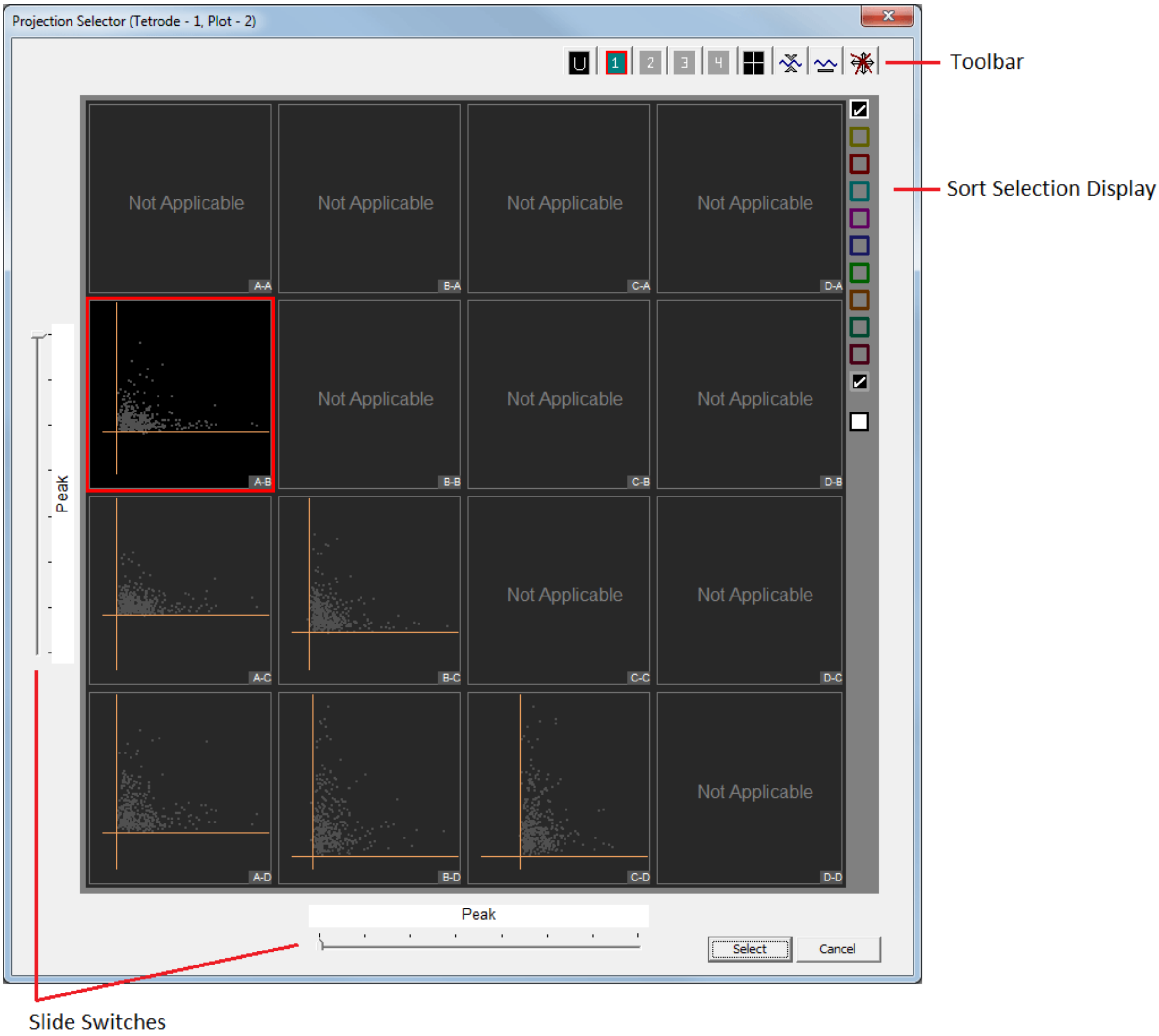
Up to four 2D projections can be used to visualize tetrode spike clustering. Each new projection can help to further refine a sort or identify new sorts. You can preview and choose projections using the active tetrode data.

Available Metrics

All metric calculations are performed on the segment of data within the min-max interval only.

Metric Name	Description
Peak	The highest data point in the interval.
Valley	The lowest data point in the interval.
Height	The difference between peak and valley in the interval.
Energy	The arithmetic mean (average) of the squares of each point in the interval.
Non-linear Energy	`sum(w(t)*w(t) - w(t-1)*w(t+1))/length(w)`, for all t in the interval, where w is the waveform array.
Average	The arithmetic mean of all values in the interval.
Area	The sum of the absolute values of all points in the interval.
Slope	The height divided by the difference between the peak timestamp and valley timestamp in the interval.

To open the Projection Selector, double-click a projection subplot. To add a projection, select a subplot that has not been configured.



Projection Selector Window

All possible combinations of metrics between channels are displayed. The letters in the lower-right corner of each plot indicate the x-axis and y-axis channels for that plot. Metric combinations that are already in use in other projection plots for this tetrode will have a solid border around them.

Use the slide switches on the left and bottom edges to chose the Y- and X-axis metrics, respectively.

The Sort Selection Display can be used to toggle the display of individual or all sort codes.

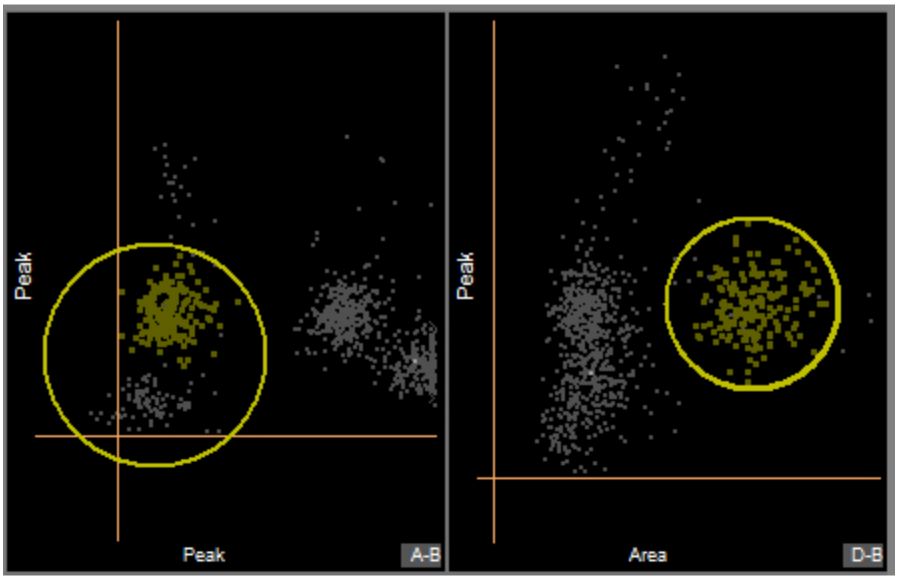
Icon	Description
	Shows all data as unsorted events
	Switch to corresponding projection (only available in plots that are already configured)
	Switch to selected cluster projection preview.
	Auto scale projections
	Reset base scale for projections
	Reset independent plot scaling removes any x-axis or y-axis skew in all available projections.

Defining Sorts

You can assign sort codes to their associated snippets by drawing a circle around the desired cluster of points in the plot. Snippets falling inside a circle are given a sort code corresponding to that circle's color. The color of the dot representing that snippet will change to the color of the circle. More than one circle of the same color can be defined in each projection. A snippet falling in any of those same-colored circles will be classified with that sort code.

Because snippets can fall into more than one circle, the sort code assigned to candidate waveforms can be either (a) the largest value of all circles the dot fell inside or (b) a mask of all sort codes that the candidate snippet fell inside. For example, if a snippet falls into a yellow (sort code 1) and green (sort code 6) circle, then the snippet mask will be 33 (`0b100001 = 33`). This assignment option is selected by the user in the Sorting Options tab at designtime.

A snippet that doesn't fall inside any circles is considered unsorted and has sort code 0 (gray). If there are circles of the same color on more than one projection, a candidate snippet has to fall inside that type of circle in all projections to be given that color's sort code. The total number of circles that can be defined in any one projection plot is set in the configuration options (the default is 12).



Sorting Interface

Drawing Circles

The circles you draw in each projection plot determine a cluster's boundaries and shape. Sort codes are applied to snippets using the boundary calculated for each cluster. Hold down the **Ctrl** key and click-and-drag to draw a sorting circle on the plot.

Sort circles can also be generated by drawing an arbitrary shape around points in a projection plot. Synapse will then attempt to draw circles that will efficiently represent the selected points. Hold down the **Alt** key and click- and-drag to draw an arbitrary shape that will be converted into sorting circles.

If necessary, the projection plot axes can be independently scaled so that the data points fit into circular clusters. To independently scale each axis, hold down **Ctrl + Alt** and click-and-drag the mouse to the left or right to scale the x-axis of the project plot, or drag up or down to scale the y-axis.

Filtering the Display by Sort Code

A column of colored squares along the right edge of the active tetrode display serves to filter events by sort code. Check the white outlined box to display all sort codes. Check the gray outlined box to display unsorted events. Hold down the **Ctrl** key and click a square, to show only that sort code.

Applying Sorts to New Data

Sort codes are not saved to the data tank until sorts are applied by the user. You can re-sort or make adjustments as needed to get the best results.

Click the **Hardware Sort** button to send the sorting parameters to the hardware and begin saving sort codes to the tank. Sort codes are applied as new data is acquired. While this button is down, changes in sorting parameters in the display will be applied automatically to new data.

Locking Channels

Click the **Lock All** button to lock the sorting circles for all channels, or right-click individual channels and choose Lock .

Click the **Unlock** button to unlock all channels, or right-click individual channel plots and choose **Unlock**.

Keyboard Shortcuts

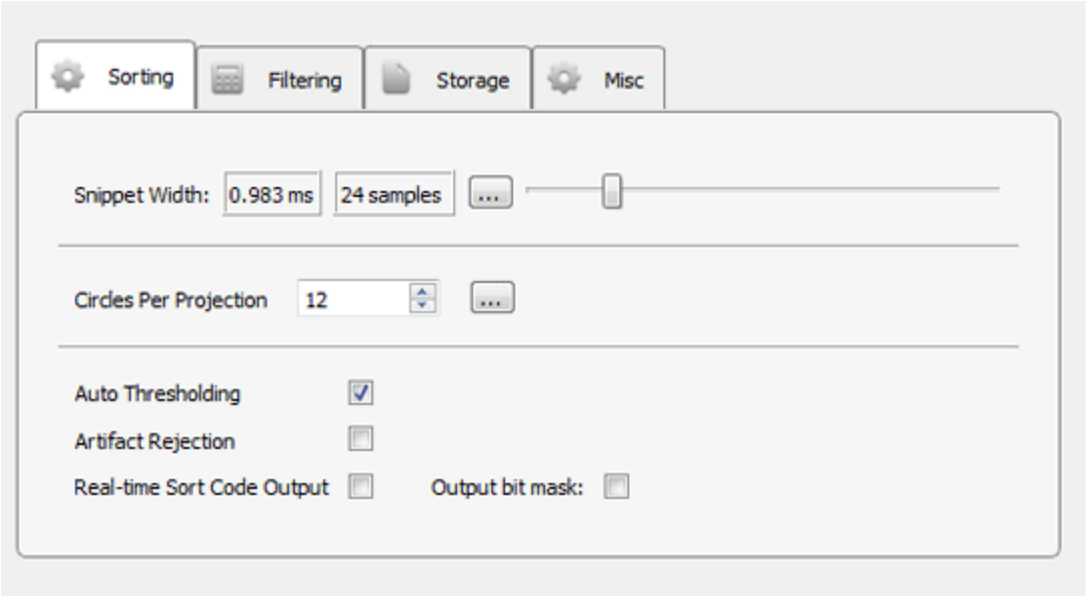
Keyboard combos	Action
Click + Drag	Pan Projection Plot.
Shift + Click and Drag	Projection Plot: Zoom in and out. Snippet Plot: Y-axis zoom.
Ctrl + Alt + Click & Drag	Skew the Projection Plots up down or left right.
Alt + Click & Drag	Snippet Plot: Pan. Projection Plot: Draw arbitrary shape.
"C" + Click Pile Plot	Send that channel to the audio monitor.

Keyboard Projection Plot hotkeys	Action
~	Show all sort codes.
1-9	Toggle sort code show/hide.
Ctrl + [1-9]	Show only the selected sort code.

Tetrode Spike Sorting Configuration Options

Sorting Tab

Settings on this tab apply to the runtime interface and snippet storage.



Sorting Options Tab

Snippet Width Slider

Drag slider to select the desired width (displayed in milliseconds and samples) of recorded snippets (per channel). The actual snippet output will be four times as long.

Circles Per Projection

Set the total number of circles that can be defined in any one projection plot. Lowering this value decreases the processing overhead.

Auto Thresholding

In automatic thresholding, the threshold used to record snippets is adjusted in real-time to changes in each channel waveform's RMS. The previous five seconds of data are used in the RMS calculation.

Artifact Rejection

When artifact rejection is enabled, snippets that contain at least one sample greater than the artifact rejection level set on the runtime interface are ignored.

Real-time Sort Code Output

Make the multi-channel integer stream of uncompressed sort codes available to other gizmos, such as Sort Binner or UDP output.

Note: The sort code output is delayed by (window width + 2) samples from when the threshold is crossed. When artifact rejection is enabled, the sort code output is delayed by an additional window width, so (2 * window width + 2) total samples.

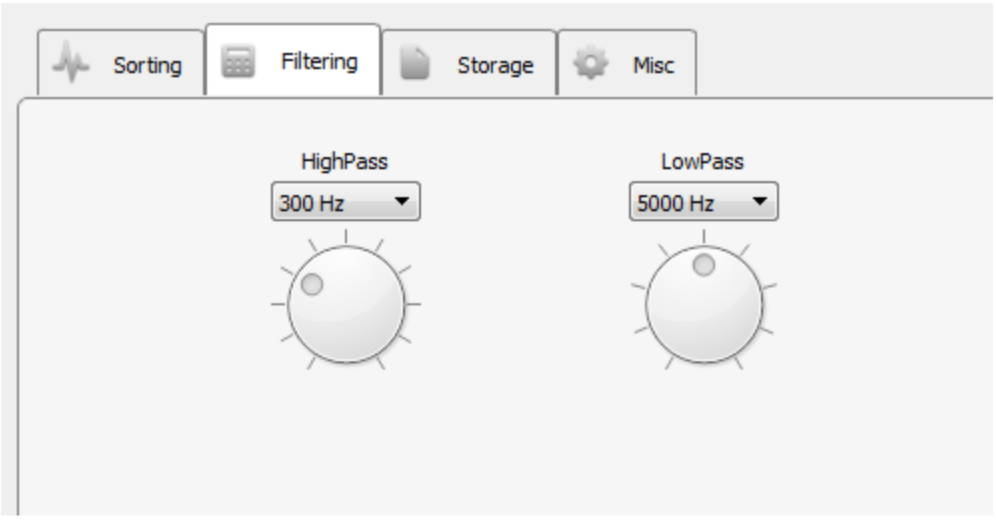
Output Bit Mask

Make the assigned sort code a mask of all sort codes that the candidate snippet fell inside. For example, if a snippet falls into a yellow (sort code 1) and green (sort code 6) circle, then the snippet mask will be 33 (`0b100001 = 33`).

The default behavior is to use the largest value of all circles the dot fell inside as the sort code. If using Sort Binner on the Sort Code output, leave this option unchecked.

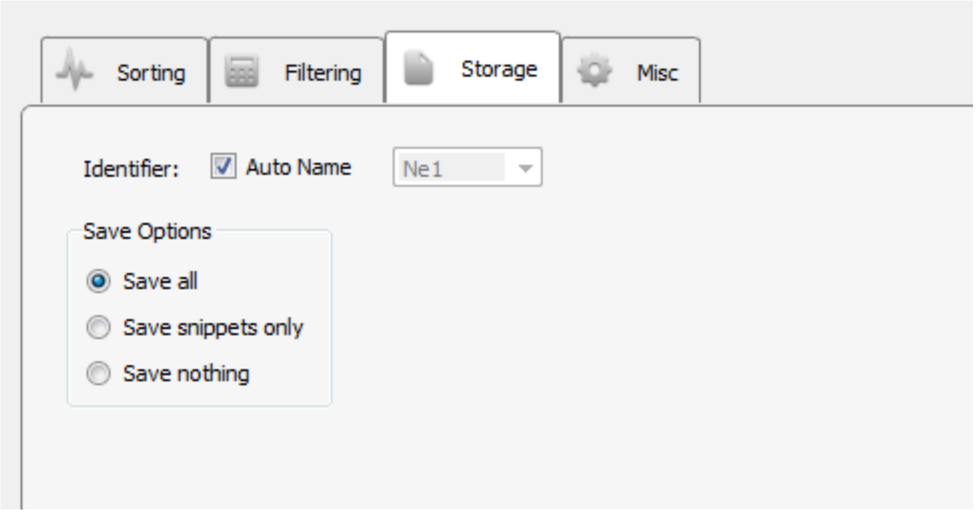
Filtering Tab

The gizmo applies a highpass and lowpass filter to all channels before spike detection. The runtime interface includes controls for dynamic adjustments to the filter settings. You also set default values in the Filtering tab.



Filtering Options Tab

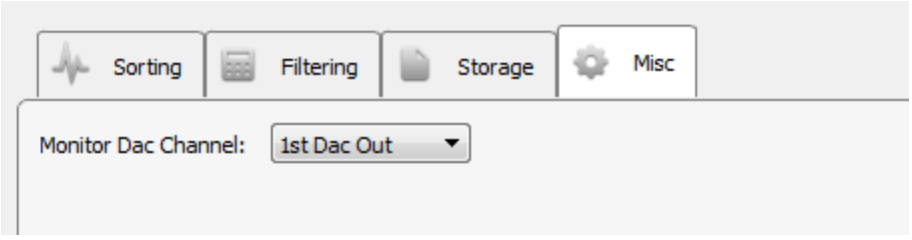
Storage Tab



Storage Options Tab

Select whether to save only snippet waveforms or to include the plot decimated waveforms used by the sorting gizmo, or to save nothing at all. The waveforms will still be displayed in the runtime interface and data plots but will not be saved to disk.

Misc Tab



Misc Options Tab

Monitor DAC Channel

Select an output channel to send the monitor signal to, or set to **Disable** to turn monitoring off.

Timer

Common Use Cases

Measures time between or duration of logical events from primary and secondary inputs. Use this gizmo to calculate event frequency or time logical events. Can be used to measure response time to stimuli, calculate heart rate, and time other physiologic intervals

Data Stored	
Epoch event (optional)	Timestamp and store the Result when a valid measurement is taken
Continuous (optional)	Store the Result output continuously
Outputs	
Result	Single channel floating point, the outcome of the timer measurement
Valid	Logic signal that triggers when the measurement is taken

Gizmo Help Slides

Quick Help Slide 1

Timer — Measure time between or duration of logical events from primary and secondary inputs

Monitor heart rate

Count the time LFP power is above a set threshold

Measure the delay between two neuronal spikes

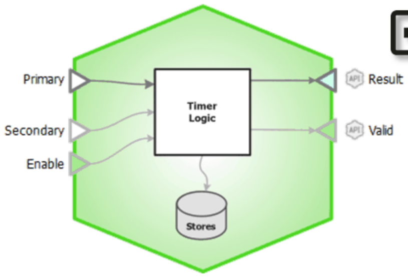
Stimulate if a spike ISI falls below a set level

Thresholding a filtered amplifier data signal to monitor heart rate in BPM

Primary: Input to measure time against

Secondary: Second signal to test Primary against

Enable: Activate gizmo. Connect to #Enable processor to always stay active



Result: Output for period, frequency, or BPM

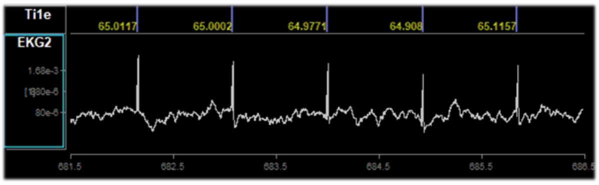
Valid: High when logic tests for primary or primary and secondary signals are true

Quick Help Slide 2

Timer

User Interface, API, Data Stored

Manual adjust threshold levels for primary and secondary input signals

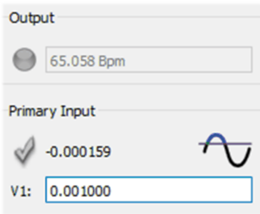


MeasCount — The total number of instances Timer has output a value {Read}

MeasOutput — Output value of timer {Read}

**Vin* — Primary or secondary input signals {Read}

**V1* — Threshold level for primary or secondary input signals {Read/Write}



Tim1 — Timer output as a continuous signal. Useful when using Hot Tracking to monitor rate changes in real-time {Stream}

Ti1e — Store the timer output value when the logic test is true {Epoch}

Reference

The Timer gizmo accepts all single channel signal types. The input first passes through a truth test. For a logic signal, this is simply a true/false test. If an integer or floating point input is used, the test can be that the signal is above/below a certain threshold, or inside/outside a range of values.

The Runtime Interface

Runtime Plot

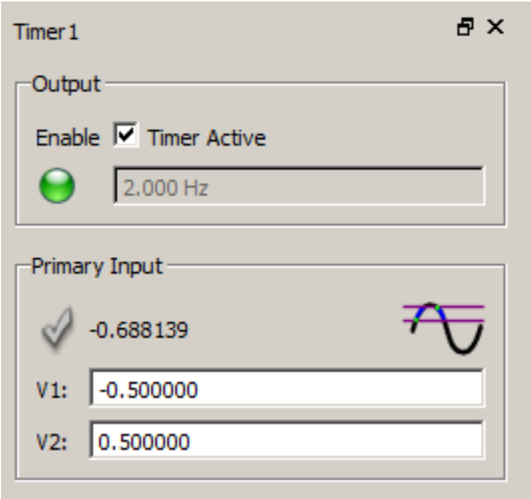
At runtime, the standard Synapse data plot displays any stored data. The timer can save timestamped epoch events when the measurements are taken, and/or a continuous stream of the measurement results.

Timer Tab

The runtime interface must be enabled in the General Tab before it can be used.

If a signal input is integer or floating point type, the values used for the truth test are adjustable at runtime.

If Enable Control is turned on, a check box on the UI shows if the timer is enabled or not.



Timer Runtime UI

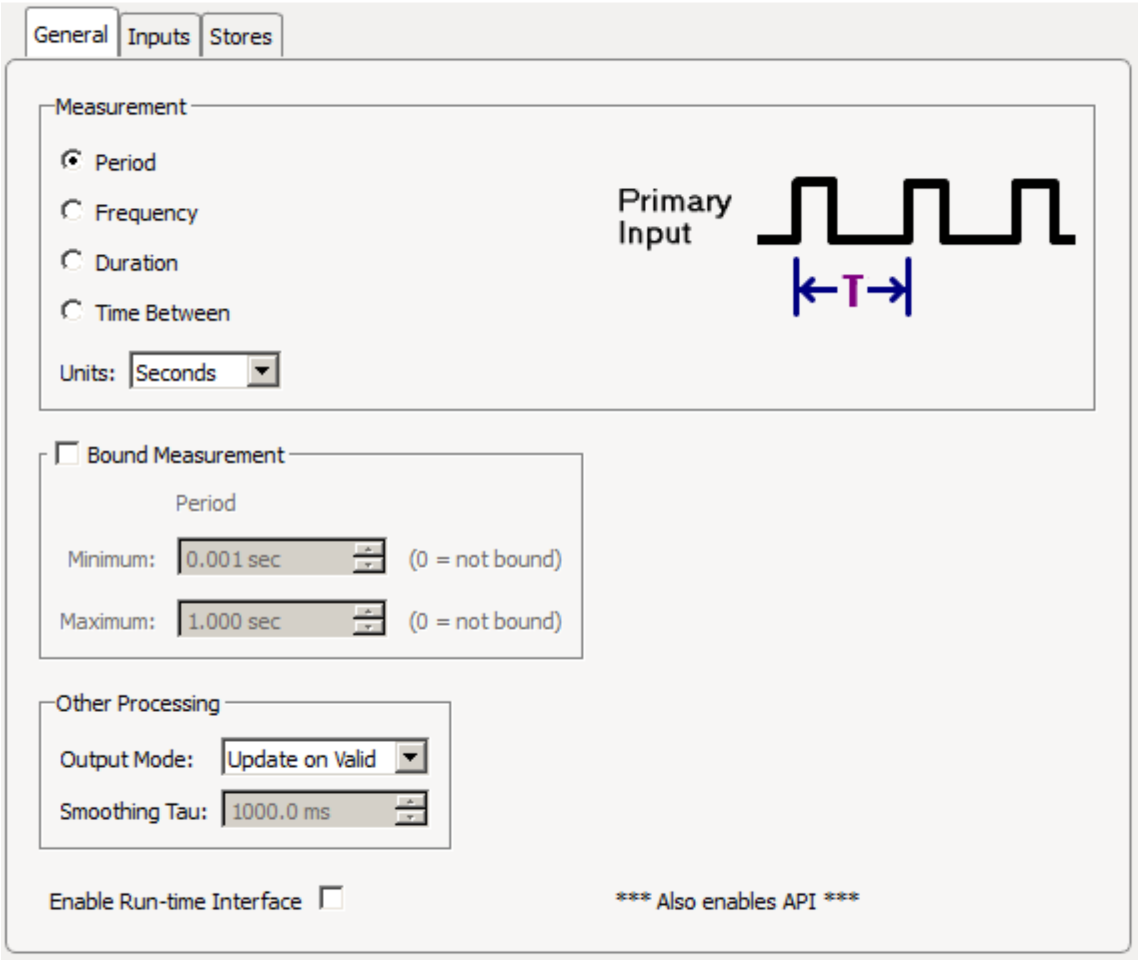
In the above example, the Enable Control option is set to 'Manual/API'. The input signal is floating point with the truth test set to 'Between'. The signal will be true if it is between -0.5 and 0.5.

The value in the Output box is the most recent measurement. The green LED is active to indicate that a measurement recently took place.

The check mark for Primary Input is not lit because the test condition is not met; the current signal value is outside the bounds.

Timer Configuration Options

General Tab

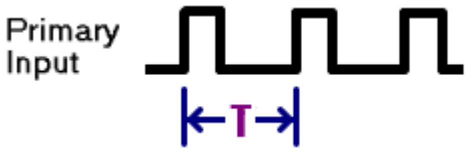


General Tab

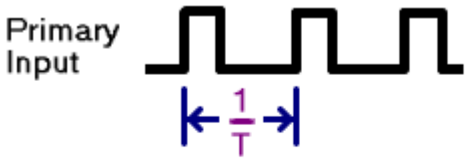
Measurement

Choose which measurement to make and the output units. The smallest possible measurement is two samples of the system clock.

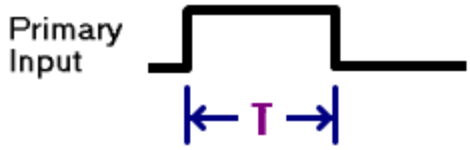
Period measures the time between consecutive onsets of the primary input signal in units of seconds, milliseconds, microseconds, or samples.



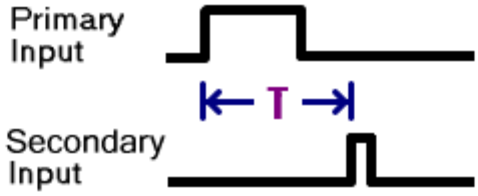
Frequency measures the frequency of consecutive onsets of the primary input signal in units of hertz, kilohertz, or beats per minute (BPM).



Duration measures the time between the onset and offset of the primary input signal in units of seconds, milliseconds, microseconds, or samples.



Time Between measures the time between the onset of the primary input signal and the onset of the secondary input signal in units of seconds, milliseconds, microseconds, or samples.



Bound Measurement

If **Bound Measurement** is enabled, the calculated measurement result will never be outside of the chosen minimum and maximum values.

In **Period** or **Frequency** measurement mode, if the measurement is outside of the limits, it will clamp to the nearest limit.

In **Duration** measurement mode, if the maximum bound is exceeded or the minimum bound isn't met, then this will not count and a measurement will not be made.

In **Time Between** measurement mode, when bounding is used the primary input is ignored when the timer is running and less than the minimum bound, so it can't re-trigger. Also, the secondary input is ignored before the minimum bound is reached and after the maximum bound is reached, so valid measurements are only taken in between the given bounds.

Other Processing

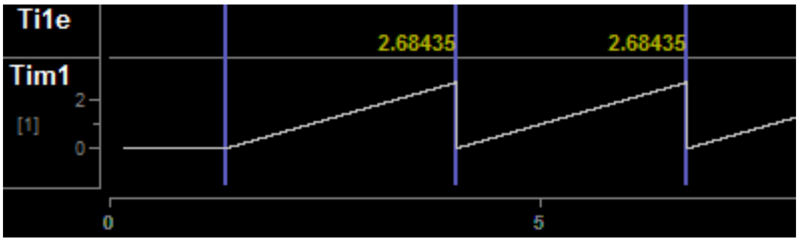
Output Mode controls how the measurement signal (Result) is handled.

Update On Valid means the Result is latched when the measurement occurs and Result holds that value until the next measurement is made.

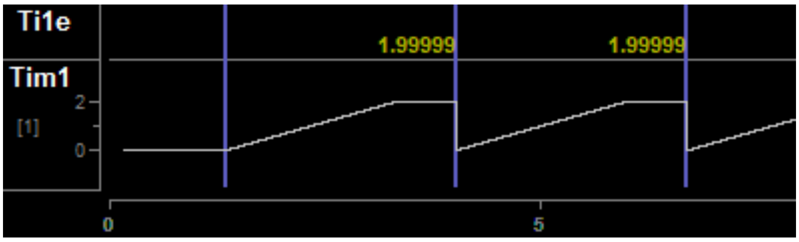
Smoothed behaves like **Update On Valid** but also applies an exponential smoothing filter with the desired **Smoothing Tau**. The larger the **Smoothing Tau**, the smoother the Result signal.

Hot Tracking provides a more instantaneous approximation of the Result while the measurement is occurring. This is useful for tracking irregular waveforms like spike firing rate.

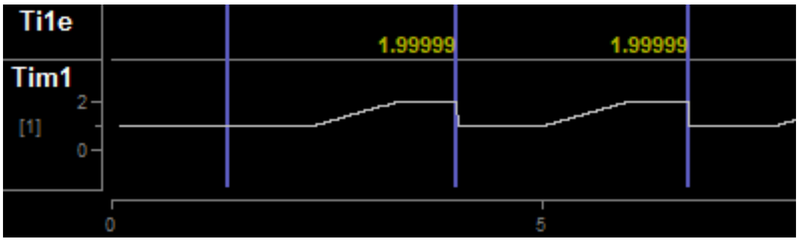
For **Period** measurements, the Result resets to 0 when a new Primary Input onset occurs and rises linearly until the next Primary Input onset, at which point a measurement is taken and it resets again.



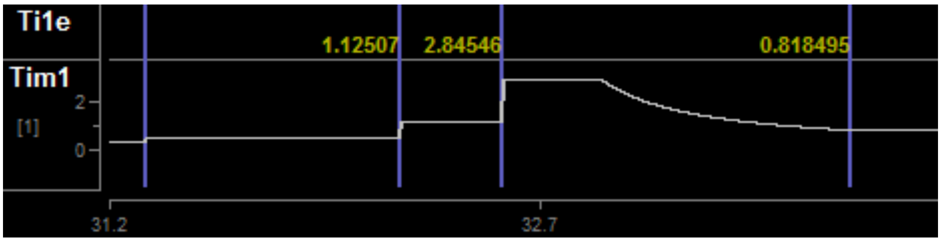
If the maximum bound is reached, Result latches at the maximum bound value until the next onset/ measurement.



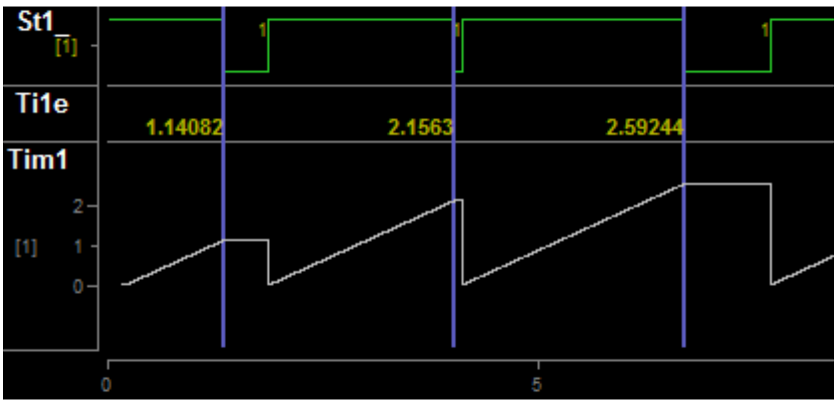
If the minimum bound is greater than zero, the Result will reset to this value (instead of 0) and will stay there until enough time has elapsed to meet the minimum bound requirement, and will then rise linearly until the next onset (or the maximum bound is reached).

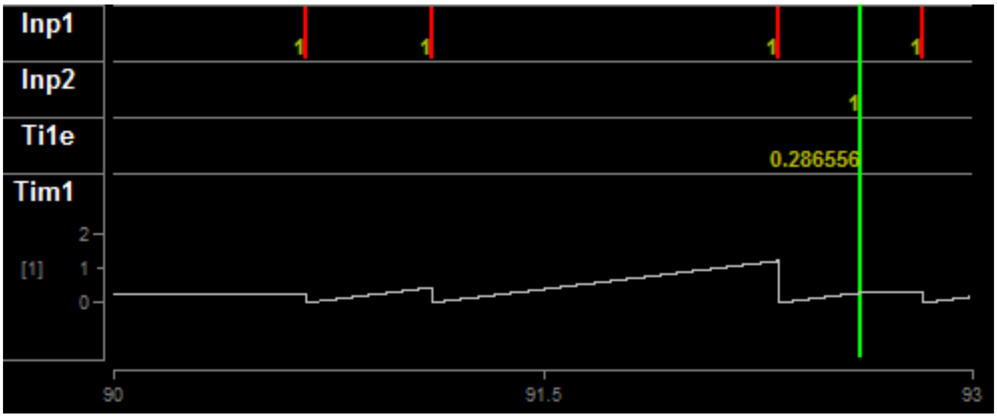


For **Frequency** measurements, the Result value latches until enough time has elapsed between onsets such that the next frequency measurement must be lower than the previous Result, at which point the Result begins decreasing in real- time until the next onset occurs (or the minimum bound is reached) and latches the new measurement value. If the frequency between onsets increases, the Result will immediately increase to the new value.



For **Duration**, the Result resets to 0 when an onset occurs and rises linearly until the offset occurs, or until the maximum bound is reached, and latches this value until the next onset.

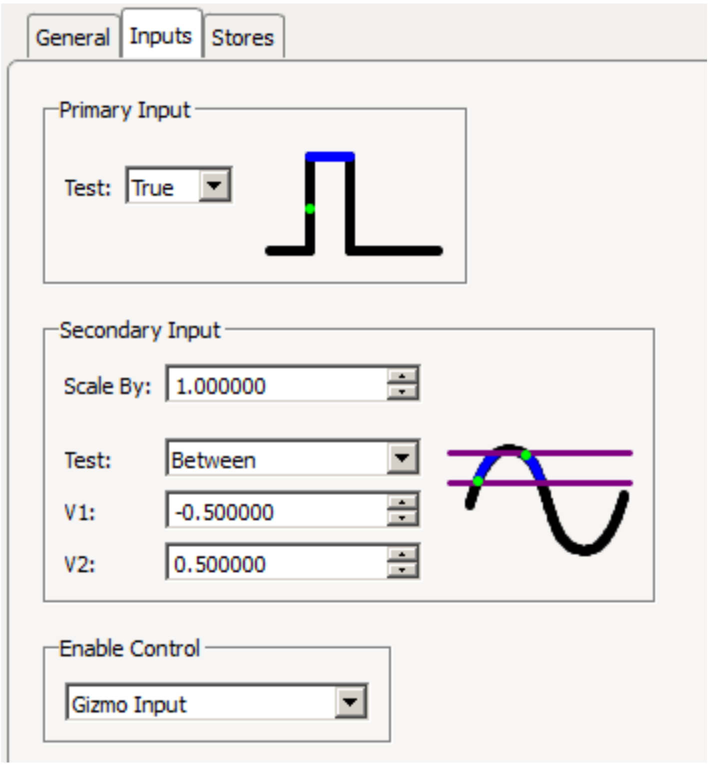




For **Time Between**, the output resets to 0 when the Primary Input onset occurs and rises linearly until the Secondary Input offsets occurs, or until the maximum bound is reached, and latches this value until the next Primary Input onset.

Inputs Tab

Select the truth tests for the input signal(s) and determine when to activate the timer processing and storage.



Inputs Tab

Primary Input

If the primary input is a Logic signal, the test can either be True or False. If the primary input is an Integer or Floating Point signal, the input can first be scaled and then a threshold Test is applied. For Above and Below, V1 is the only value shown and used for the threshold test. For Between and Outside, V1 is the minimum bound and V2 is the maximum bound.

Secondary Input

The Secondary Input is only visible if the Measurement type is **Time Between**. This contains the same truth test options as the Primary Input above.

Enable Control

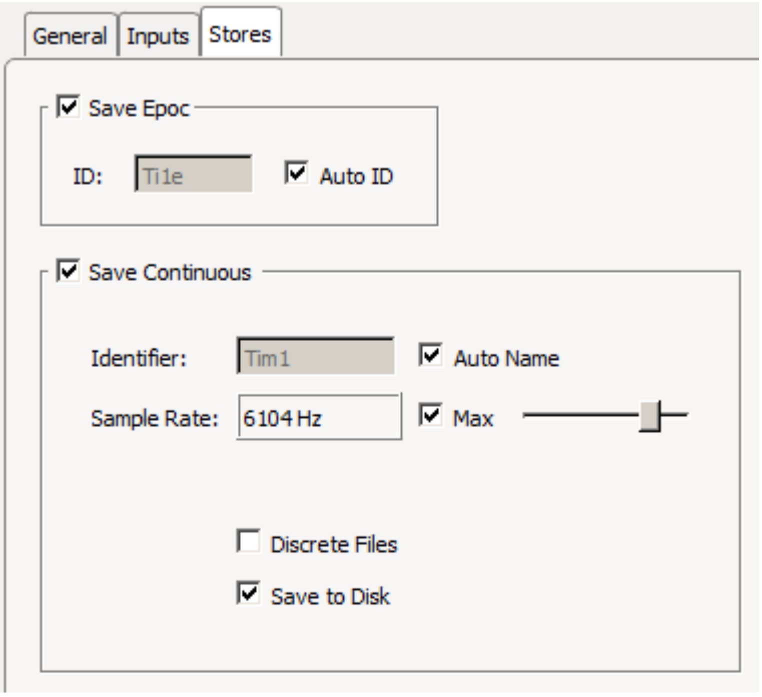
'None' means the timer is always active.

'Gizmo Input' means the timer is only active when an additional gizmo input signal (a Logic signal called 'Enable') is True.

'Manual/API' enables user control of timer processing. Note that this is only available if the Run-Time Interface is Enabled on the Misc Tab.

Stores Tab

Choose what data (if any) to store.



Stores Tab

Save Epoc

Store the timestamp and Result when Valid is true (when a measurement is taken).

Save Continuous

Saves the Result continuously and makes it visible on the runtime Data Plot.

Ultrasonic File Stimulation

Common Use Cases

Play custom waveforms at ultrasonic frequencies (200 kHz sampling rate) from a list of files on disk, which includes WAV files and MAT files. Use for audio neurophysiology and stimulus response protocols for animals that can hear in the ultrasonic frequency range.

Data Stored	
Epoc (optional)	Parameter values when triggered
Stream (optional)	Raw stimulus waveform
Outputs	
Stim	Stimulus waveform, single channel floating point
StimSync	Logic signal when file stimulation is active
Par Output	Full parameter stream

Gizmo Help Slides

Quick Help Slide 1

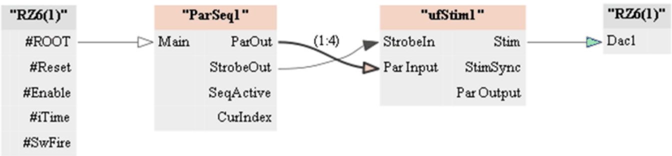
Ultrasonic File Stimulation — Play custom waveforms from a list of files on disk, which includes WAV, F32, and MAT files. Optimized for high frequency stimulation

Present custom auditory stimulation patterns in ultrasonic frequency range

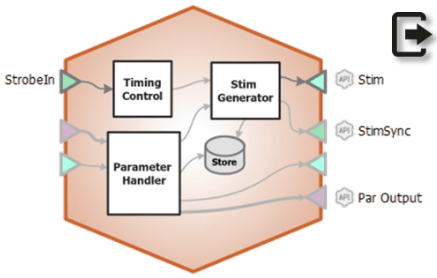
Social experiments with mouse and rat vocalizations

High-frequency ABRs

RZ6 processor uses Parameter Sequencer to control ufStim1 gizmo for stimulus presentation. Signal sent to DAC1 on RZ6 for play out of a speaker.



StrobeIn: Optional input to control presentation timing



Stim: Single-channel stimulation signal

StimSync: Logic signal that is high during a stimulus presentation

ParOut: Parameter values for the given presentation

Quick Help Slide 2

Ultrasonic File Stimulation

User Interface, API, Data Stored

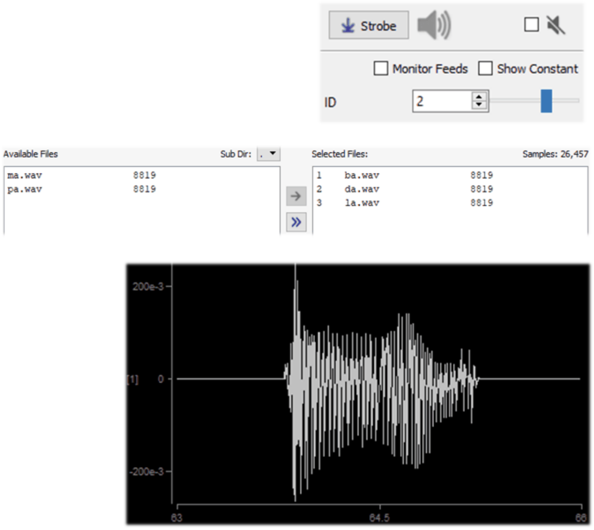
Manually strobe and mute stimuli presentations. Change file ID during runtime

Mute — Turn the stimulus on & off during a presentation {Read/Write}

Strobe — In Manual mode, strobe the gizmo to give a stimulus presentation {Write}

StimActive — Logic high during a stimulus presentation {Read}

uf1r — Raw output waveform {Stream}



Quick Help Slide 3

Ultrasonic File Stimulation

Parameter Table

This gizmo has a Parameter Table to dynamically control values its parameters from other gizmos and during runtime

Parameter Tables provide the user dynamic control over parameter values. Each one of the Modes will change how the user can control and interact with the parameter.

Param In – Dynamically control parameter values from Parameter Sequencer or Parameter Manifold

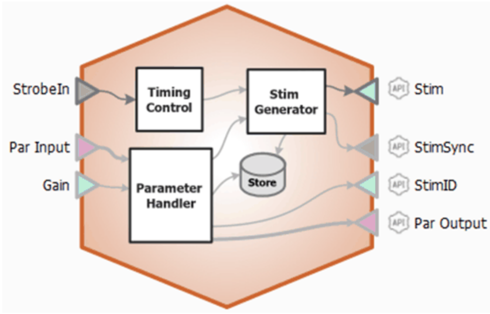
Constant – The value is immutable at runtime

Scalar In – Pass a scalar value from another gizmo for dynamic real-time control

Widget – Creates a widget (slider) on the user interface that is controlled manually by the user or through the API.

Scalar Out (Scout) – Makes this value available as a gizmo output to connect to another gizmo input in real-time

Epoc – Save the parameter values used for each stimulus presentation



Changing the Mode of parameters will change the input and output options on the gizmo.

All parameters can be read through API. Only parameters in Widget mode can be written to.

All parameters are connected to the "Par Output" gizmo output link

File Stim Parameters: Show All

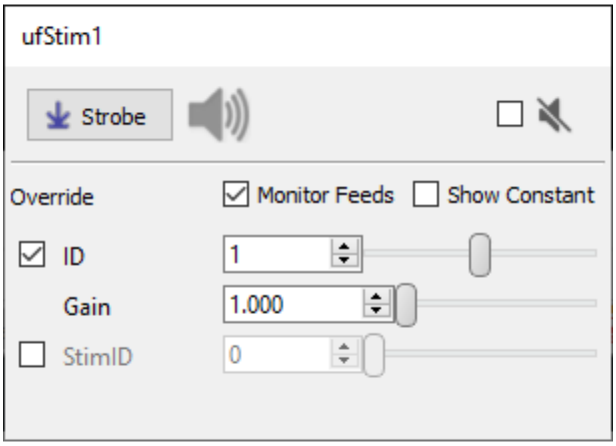
	Name	Mode	Value	Min	Max	Epoc	ID	Auto ID	SCout-1
1	ID	Param In	1	0	2	None	WVid	☑	○
2	Gain	Scalar In-1	1.000	0.000	1000000.0(	None	WVgn	☑	○
3	StimID	Param In	0	0	10000	None	STid	☑	○

Reference

The Ultrasonic File Stimulation gizmo plays stimulus waveforms from a list of files. It supports timing control and dynamic parameters.

File Stimulation Runtime Interface

A control tab is optionally added at runtime. Parameters that can be controlled dynamically are shown in black (active).



Example of Ultrasonic File Stimulation Runtime Tab

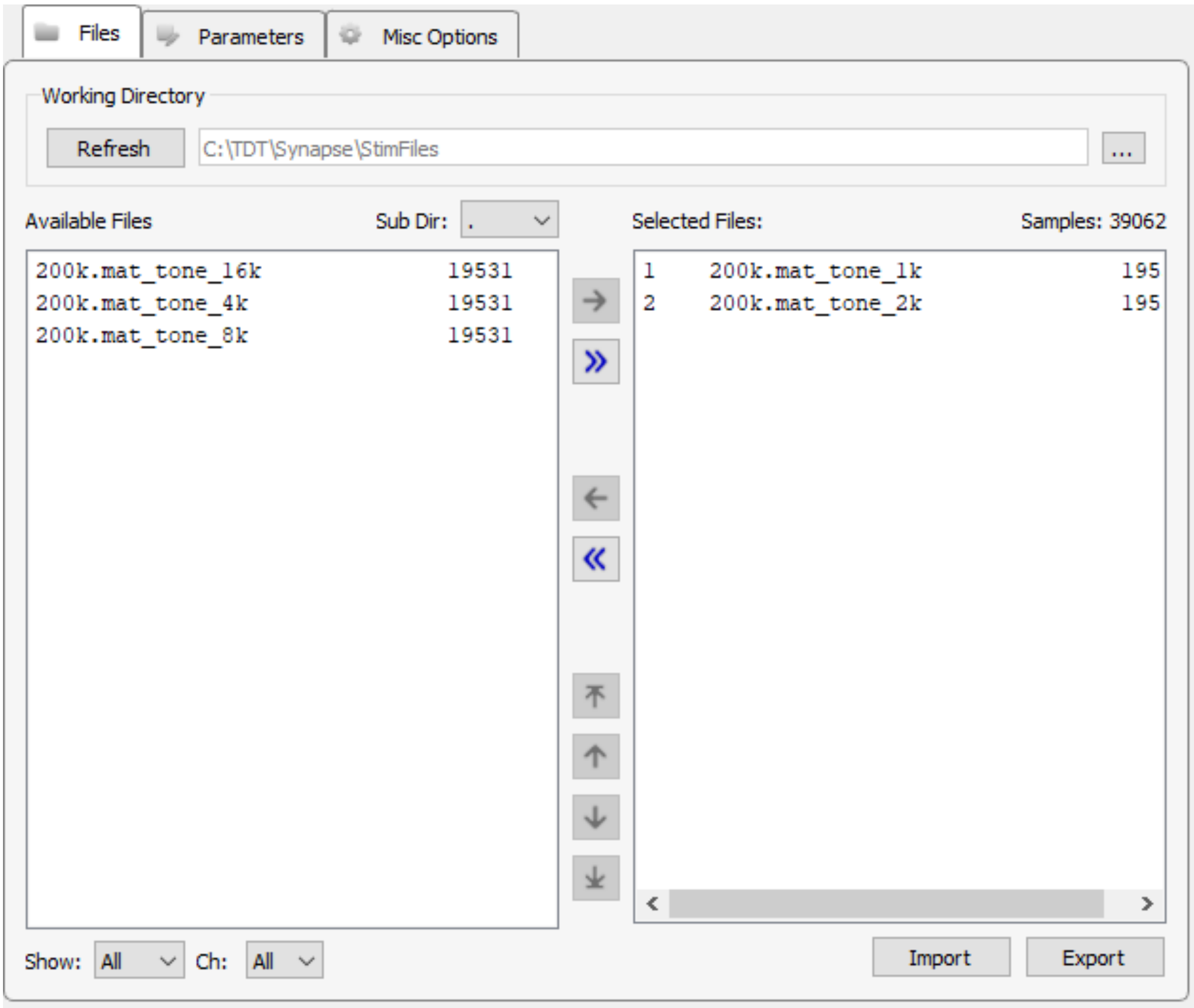
Click and release the **Strobe Button** to trigger a manual strobe pulse.

Select the **Mute Button** check box to zero the stimulus signal.

Select the **Monitor Feeds** check box to show stimulus parameters controlled by an input signal. This also adds an Override column and check box to the left that lets you adjust the parameter value manually instead of using the input signal.

Ultrasonic File Stimulation Configuration Options

Files Tab



Files Tab

Working Directory

The default working directory is C:\TDT\Synapse\StimFiles. You can select a different directory or stick with the default. If you add files to the directory or choose a new directory, you can click **Refresh** to update the displayed list of available files below.

The lower portion of the window serves as a simple graphical interface for displaying, filtering, and selecting stimulus files for play out.

Available Files

In the list on the left, all stimulus files found in the working directory are displayed. Stimulus files can be any of the following types:

- continuous 32-bit floating points (*.f32)
- continuous 32-bit integers (*.i32)

- continuous 16-bit integers (*.i16)
- Wave (*.wav)
- MATLAB arrays (*.mat)
- text file with one value per line (*.txt)
- text file with comma-separated values (*.csv)

A **Show Types** drop-down filter, below the **Available Files** area, narrows the displayed files to the selected file type. The **Sub Directory** drop-down menu allows you to drill down to subdirectories within the working directory.

Selected Files

The area to the right, serves as a list of files to be loaded as the stimuli.

File Buttons

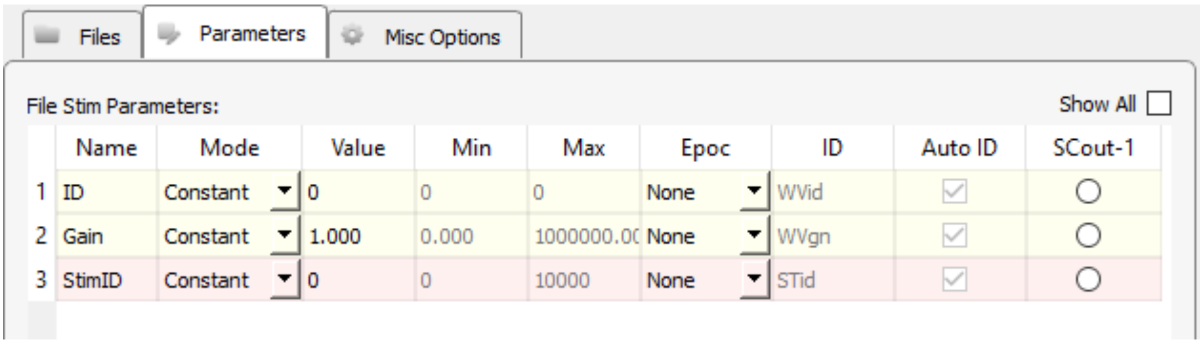
Use the file buttons, located between the two lists, to choose the files to use.

Button	Description
	move a file from available to selected
	move all files to Selected Files
	move a file from selected to available
	move all files to Available Files
	move file to top of list
	move file up in list
	move file down in list
	move file to the bottom of the list

Import /Export

These buttons can be used to import or export stimulus files.

Parameters Tab



Parameter Tab

File Stim Parameters

The table lists parameters relevant to configuring the stimulus. Use the **Show All** check box to display hidden rows.

> **Tip**
>
> See Using Parameters for more information on controlling parameter tables.

The ID parameter is the current file ID to present from the Selected Files list in the Files tab. The StimID is a custom stim marker that you can use for organizing data for easier analysis. In a typical averaging example you might present the same file ID but alternate the gain between 1 and -1 to remove stimulus artifacts. For analysis purposes you want those combinations of parameters to have the same StimID for easier extraction and averaging.

Mode

In the Mode column, you can choose to make individual parameter Constant, controlled by a runtime Widget, or controlled by an Input line.

Value Columns

Enter values in the Value, Min, and Max columns to set the Constant value or to set the initial value when a Widget control will be used.

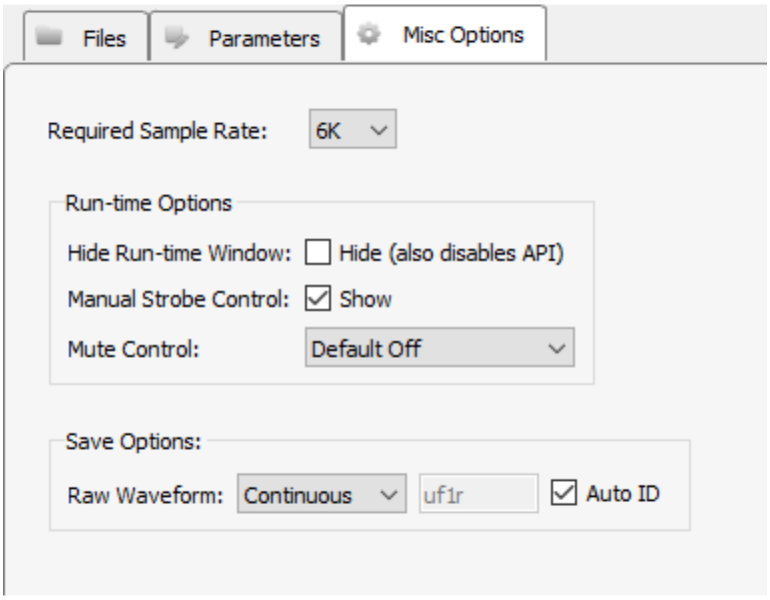
Epoc

In the Epoc column, you can choose to save the individual parameter value on stimulus onset.

ID and Auto ID check box

Synapse automatically generates a store name. TDT recommends using Auto ID to ensure no store names are duplicated. A "/" is appended to the name to indicate when the full epoc is stored (and is not appended when only saving the onset). To make your own store names, clear the **Auto ID** check box.

Misc Options Tab



Misc Options Tab

Required Sample Rate

The minimum rate required. Synapse looks through the entire system and sets the sample rate according to this and other limiting factors.

Hide Run-Time Windows check box

By default a runtime tab is added in preview or record mode. The contents of the tab are defined with configuration options on the General and Parameter options tab. Select the check box to hide the runtime tab.

Manual Strobe Control check box

When selected a manual strobe control is added to the runtime eStim tab. Clear the check box to hide the manual strobe control at runtime.

Mute Control

Select the default behavior of the runtime mute control. Mute allows you to mute or temporarily zero the stimulus during runtime. You can choose to hide or show the control and, if show, set the default start state.

Raw Waveform

Select whether to store a copy of the raw stimulus waveform. You can choose to store continuously or only when the stimulus is active.

Ultrasonic Stimulation

Common Use Cases

A streamlined version of the Audio Stimulation gizmo for creating stimuli at ultrasonic frequencies. This gizmo is useful for audio neurophysiology and stimulus response protocols for animals that can hear in the ultrasonic frequency range.

Data Stored	
Epoc (optional)	Parameter values when triggered
Stream (optional)	Raw stimulus waveform
Outputs	
Stim	Stimulus waveform, single channel floating point
StimSync	Logic signal when file stimulation is active
Par Output	Full parameter stream

Gizmo Help Slides

Quick Help Slide 1

Ultrasonic Stimulation — An optimized version of Audio Stimulation Gizmo for creating stimuli at ultrasonic frequencies

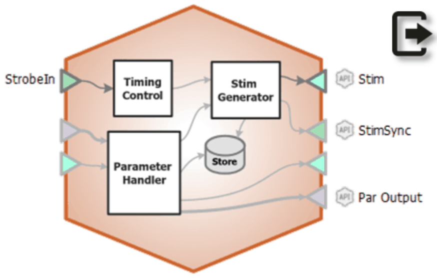
Social experiments with mouse and rat vocalization cues

Intracranial ultrasonic neuron stimulation

High-frequency ABRs

RZ6 processor #Reset tied to the uStim1 gizmo allows manual control of stimulus presentation. Signal sent to DAC1 on RZ6 for play out of a speaker.

StrobeIn: Optional input to control presentation timing



Stim: Single-channel stimulation signal

StimSync: Logic signal that is high during a stimulus presentation

ParOut: Parameter values for the given presentation

Quick Help Slide 2

Ultrasonic Stimulation

User Interface, API, Data Stored

Manually strobe and mute stimuli presentations. Change waveform parameters during runtime

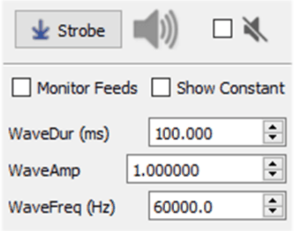
Mute — Turn the stimulus on & off during a presentation {Read/Write}

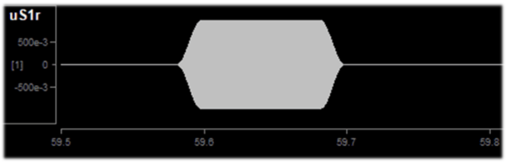
ParameterList — Array of all stimulus parameters {Read}

StimActive — Logic high during a stimulus presentation {Read}

Strobe — In Manual mode, strobe the gizmo to give a stimulus presentation {Write}

uS1r — Continuous audio output stream {Stream}





Quick Help Slide 3

Ultrasonic Stimulation

Parameter Table

This gizmo has a Parameter Table to dynamically control values its parameters from other gizmos and during runtime

Parameter Tables provide the user dynamic control over parameter values. Each one of the Modes will change how the user can control and interact with the parameter.

Param In – Dynamically control parameter values from Parameter Sequencer or Parameter Manifold

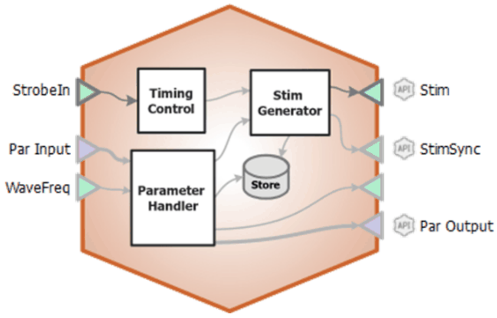
Constant – The value is immutable at runtime

Scalar In – Pass a scalar value from another gizmo for dynamic real-time control

Widget – Creates a widget (slider) on the user interface that is controlled manually by the user or through the API.

Scalar Out (Scout) – Makes this value available as a gizmo output to connect to another gizmo input in real-time

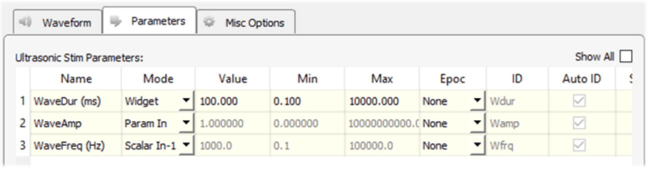
Epoc – Save the parameter values used for each stimulus presentation



Changing the Mode of parameters will change the input and output options on the gizmo.

All parameters can be read through API. Only parameters in Widget mode can be written to.

All parameters are connected to the "Par Output" gizmo output link



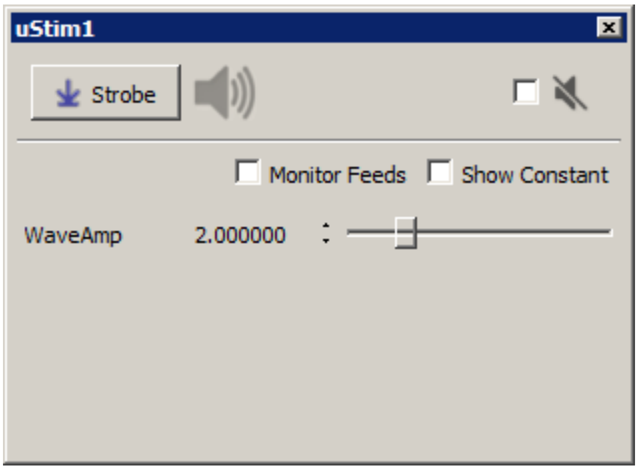
	Name	Mode	Value	Min	Max	Epoc	ID	Auto ID
1	WaveDur (ms)	Widget	100.000	0.100	10000.000	None	Wdur	☑
2	WaveAmp	Param In	1.000000	0.000000	10000000000.(	None	Wamp	☑
3	WaveFreq (Hz)	Scalar In-1	1000.0	0.1	100000.0	None	Wfrq	☑

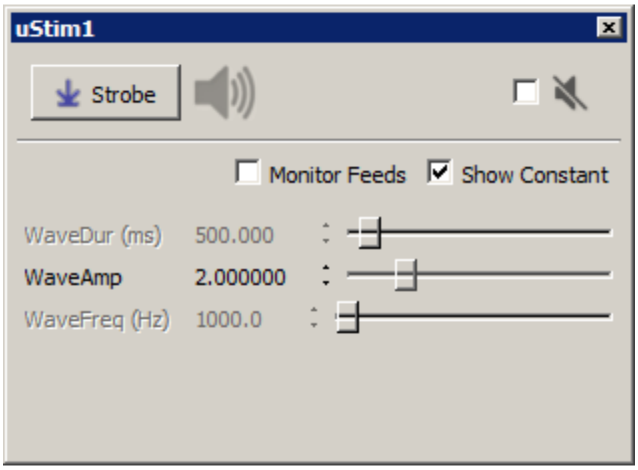
Reference

The Ultrasonic Stimulation gizmo configures timing, parameter handling, and audio stimulation generation. It is a simplified version of the Audio Stimulation gizmo that is capable of stimulating up to 85 kHz pure tones and Gaussian noise on the RZ6 processor.

Ultrasonic stimulation waveforms may be comprised of tones or Gaussian noise that can vary in duration, level, and frequency. The gizmo provides static or runtime control of stimulus parameters and can input parameters from a Parameter Sequencer gizmo. The audio stimulation gizmo includes options to store individual parameters and raw waveform. An output timing pulse can synchronize data collection.

Ultrasonic Stimulation Runtime Interface





Two Versions of the Ultrasonic Stimulation Runtime Tab

If enabled in the gizmo configuration, a uStim1 control tab is added at runtime. Parameters that can be controlled dynamically are shown in black (active). You can enter a value in the field, use up and down arrows, or drag a slider to modify to parameter value. You can show only the elements you need or hide the entire control. The illustrations above, show two version of the floated tab, one with only the runtime widget controlled parameter shown and one with all the parameters shown.

Click and release the **Strobe Button** to trigger a manual strobe pulse.

Select the **Mute Button** check box to zero the stimulus signal.

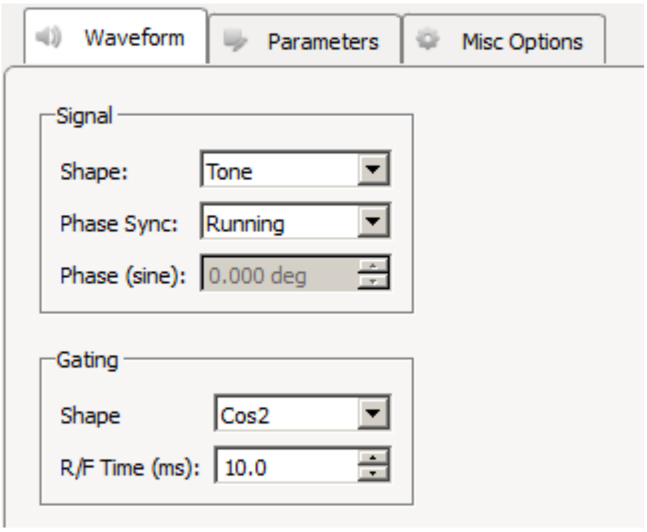
Select the **Monitor Feeds** check box to show stimulus parameters controlled by an input signal. Also adds an Override column and check box to the left. Select the **Override** check box to adjust the parameter value manually instead of using the input signal.

Select the **Show Constant** check box to display values for parameters set to Constant. They will appear gray.

Ultrasonic Stimulation Configuration Options

Use the options tabs to enable/disable optional features and set parameters that will be used to configure the gizmo operation and interface. Changes are not applied until you commit all settings. See The Options Area and Templates for more information on the gizmo name, source, global options, and displaying the block diagram.

Waveform Tab



Waveform Tab

Signal

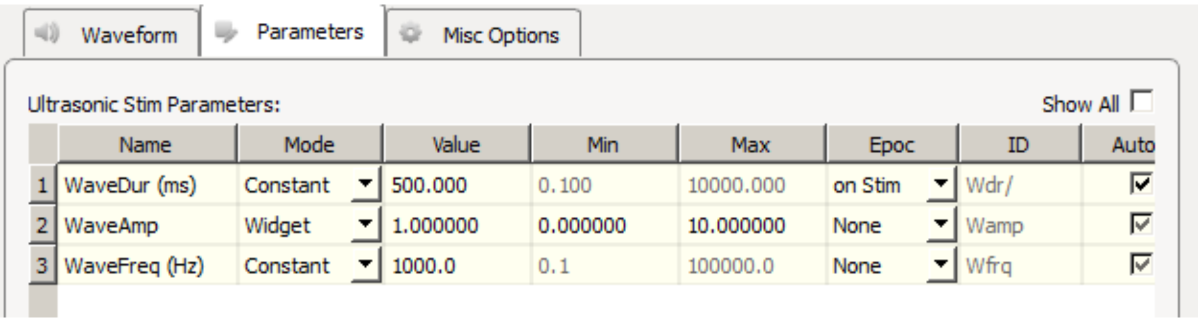
Select the desired waveform shape and whether to synchronize the phase of the waveform. For frozen noise, select the **Gauss Noise** Shape and set Phase Sync to **Sync to Stim**. Note that the Phase (sine) variable is limited to the range [-179.9, -179.9], even if it allows you to enter ±180.

Gating

Choose the **Shape** of the gate to apply to the signal. Gates serve to attenuate the signal during the onset and offset of the signal, increasing or decreasing in intensity, for the purpose of removing onset/offset related artifacts from this signal.

The **R/F Time (ms)** defines the length of time over which the gate is applied, therefore, the length of time in which the signal goes from 0 to full signal strength or visa-versa.

Parameters Tab



	Name	Mode	Value	Min	Max	Epoc	ID	Auto
1	WaveDur (ms)	Constant	500.000	0.100	10000.000	on Stim	Wdr/	☑
2	WaveAmp	Widget	1.000000	0.000000	10.000000	None	Wamp	☑
3	WaveFreq (Hz)	Constant	1000.0	0.1	100000.0	None	Wfrq	☑

Ultrasonic Stimulation Parameters Tab

Ultrasonic Stimulation Parameters

The table lists signal parameters relevant to configuring a stimulus. Each row represents a parameter and rows are shown or hidden in response to selections made on the Waveform tab. Use the **Show All** check box to display hidden rows.

> **Tip**
>
> See Using Parameters for more information on controlling parameter tables.

Mode

In the Mode column, you can choose to make individual variable Constant, controlled by a runtime Widget, fed by a parameter input line (from a Parameter Sequencer gizmo) or controlled by a Scalar Input line.

Value Columns

Enter values in the Value, Min, and Max columns to set the Constant value or to set the initial value and limits when parameters are dynamically controlled. In Widget mode, the Min and Max set the Widget limits.

Epoc

In the Epoc column, you can choose to save the individual parameter value on stimulus onset.

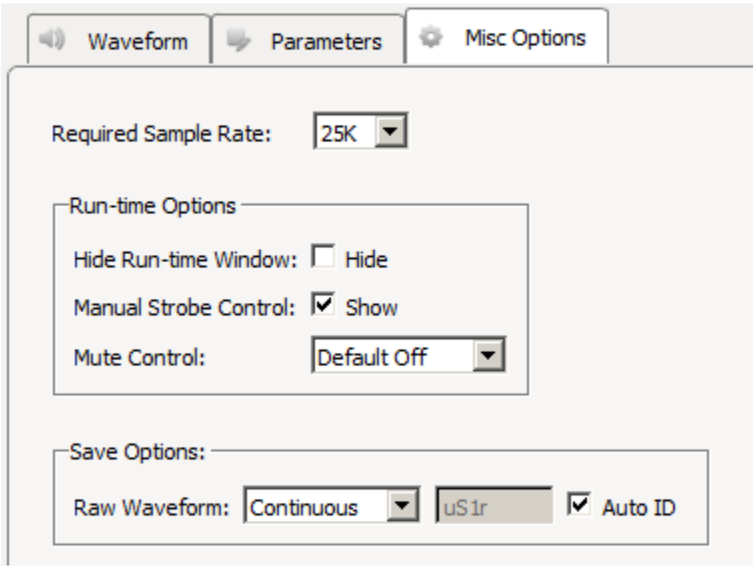
ID and Auto ID check box

Synapse automatically generates a store name for each parameter. TDT recommends using Auto ID to ensure no store names are duplicated. A "/" is appended to the name to indicate that the full epoc (onset and offset timestamp) is stored.

SCout-1

Select the radio button in the desired row to feed the parameter to an output signal on the gizmo.

Misc Options Tab



Misc Options Tab

Required Sample Rate

The minimum sampling rate required. Synapse looks through the entire experiment and your Rig and sets the sample rate according to this and other limiting factors.

Hide Run-Time Windows

By default a runtime tab is added in Preview or Record mode. The contents of the tab are defined with configuration options on the General and Parameter options tab. Select the check box to hide the runtime tab.

Manual Strobe Control

When selected, a manual strobe control is added to the runtime UI.

Mute Control

Mute allows you to temporarily mute the stimulus during runtime. You can choose to hide or show the control and, if shown, set the default start state.

Raw Waveform

Select whether to store a copy of the raw stimulus waveform. You can choose to store continuously or only when the stimulus is active.

Unary Signal Processor

Common Use Cases

Implement series of mathematical operations to incoming signals. Use this gizmo to perform interesting signal processing on incoming data, such as power in band, RMS, or scaling. Can also perform complex thresholding or type conversion on signals.

Data Stored	
Epoc (optional)	Parameter values and timestamp of changes
Outputs	
Main	Single or multi-channel floating point signal
Par Output	Full parameter stream
AltOut	Output from logic test if converting to TTL

Gizmo Help Slides

Quick Help Slide 1

Unary Processor — Apply a custom series of filtering and math to a single or multichannel signal

Calculate power in band (sleep state detection)

Convert data types

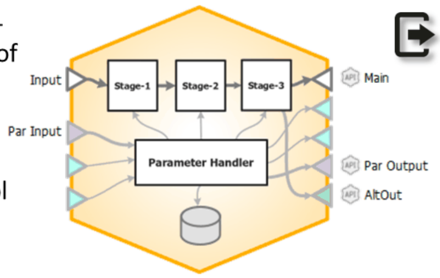
Perform complex thresholding (closed-loop stimulation off of signal power)

Cleanup and convert signals from lab sensors (nose poke sensor)

32-Channel LFP-filtered data to Unary Processor to calculate Power in Band. Result stored to disk.

Input: Single or multi-channel data stream of any type

Par Input: Input from Parameter Sequencer to dynamically control an array of gizmo parameters



Main: The processed signal

Par Output: Parameter values for all stages from the parameter table

AltOut: Output from logic test (if converting to TTL)

Quick Help Slide 2

Unary Processor

User Interface, API, Data Stored

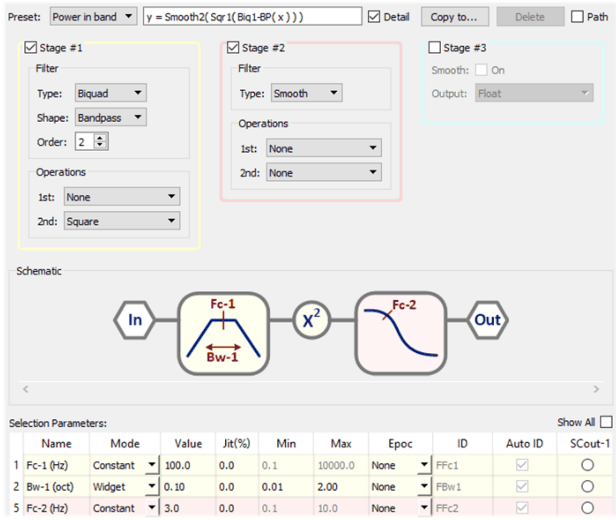
Use preset calculators such as Power in Band, Power, RMS in Band, RMS, or create your own.

API depends on which stages and options are active. Set parameter Mode to 'Widget' to enable run-time control and Write access. An example (see dialog below)

Bw-1 — Change the bandwidth of the bandpass filter (in octaves) {Read/Write}

Fc-{1…2} — Change the center frequency of bandpass filter (Fc-1) and corner frequency (Fc-2) of low pass smoothing filter {Read/Write}

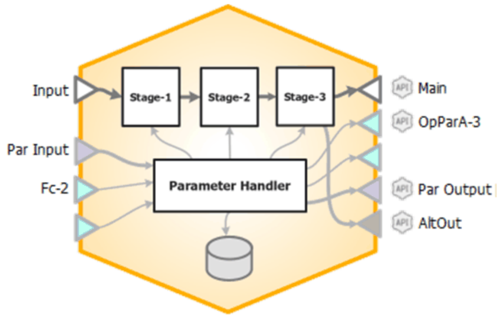
Epoc — Capture and store changes to parameter values {Epoch}



Quick Help Slide 3

Unary Processor

Parameter Table



This gizmo has a Parameter Table to dynamically control values its parameters from other gizmos and during runtime

Parameter Tables provide the user dynamic control over parameter values. Each one of the Modes will change how the user can control and interact with the parameter.

Param In – Dynamically control parameter values from Parameter Sequencer or Parameter Manifold

Constant – The value is immutable at runtime

Scalar In – Pass a scalar value from another gizmo for dynamic real-time control

Widget – Creates a widget (slider) on the user interface that is controlled manually by the user or through the API.

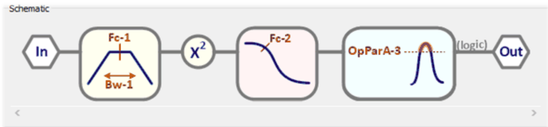
Scalar Out (Scout) – Makes this value available as a gizmo output to connect to another gizmo input in real-time

Epoc – Save the parameter values used for each stimulus presentation

Changing the Mode of parameters will change the input and output options on the gizmo.

All parameters can be read through API. Only parameters in Widget mode can be written to.

All parameters are connected to the "Par Output" gizmo output link



	Name	Mode	Value	Jit(%)	Min	Max	Epoc	ID	Auto ID	SCout-1
1	Fc-1 (Hz)	Param In	100.0	0.0	0.1	10000.0	None	FFc1	☑	○
2	Bw-1 (oct)	Constant	0.10	0.0	0.01	2.00	None	FBw1	☑	○
5	Fc-2 (Hz)	Scalar In-1	3.0	0.0	0.1	10.0	None	FFc2	☑	○
11	OpParA-3	Widget	1.000	0.0	-100000000	100000000	None	OPA3	☑	◉

Reference

The Unary Signal Processor gizmo applies a series of mathematical operations to a single or multi-channel signal. Operations such as RMS and power band calculations are available as presets. Other available operations include adding custom FIR or IIR filters, calculating absolute value, converting data types, and many others, all in one gizmo. Example uses include triggering based on power in a frequency band, or processing/converting external sensor voltages.

The gizmo uses a bounded parameter table (see Using Parameters) to define values for some operations, such as filter settings and scale/shift parameters. Optionally, you can enable runtime control or storage of these gizmo parameters.

The Runtime Interface

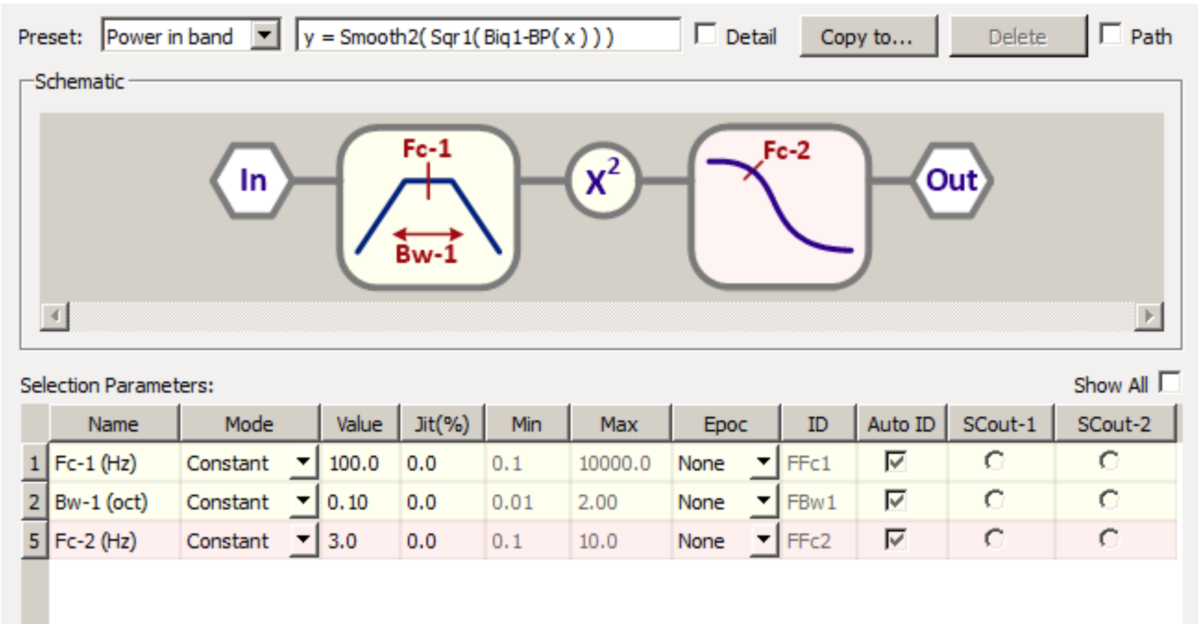
Runtime Plot

This gizmo is not specifically associated with any plotting, other than optionally storing parameter values when they change, which is not typical. If you want to view the output data during an experiment, you can add a Stream Data Storage gizmo to the output.

Runtime Parameter Controls

The gizmo parameters will be shown at runtime. You can enable runtime control of any parameter by selecting **Widget** in the parameter table. See Using Parameters for more information.

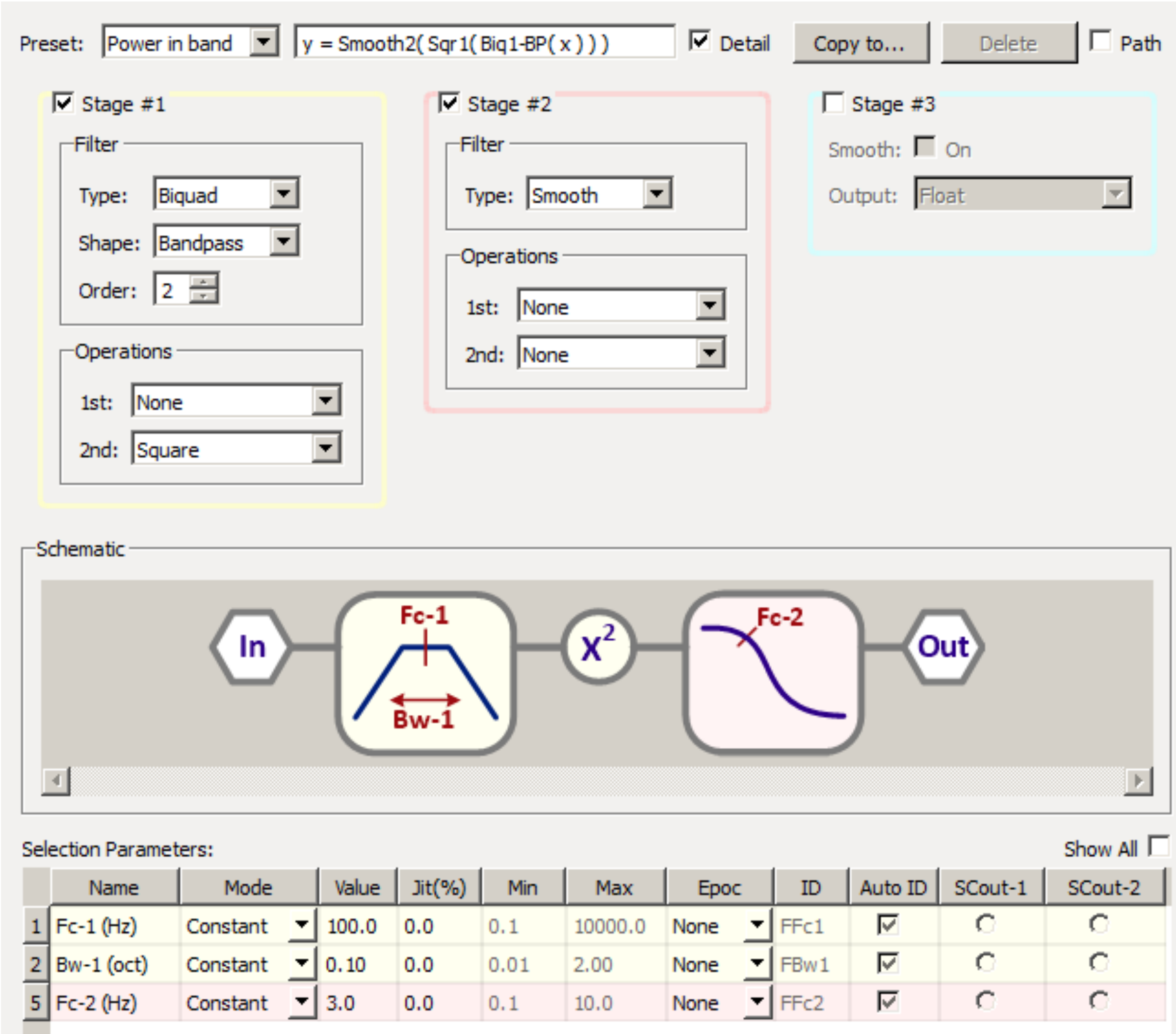
Unary Signal Processor Configuration Options



Options Area

The Unary Options area displays the mathematical formula, an image showing the operations and parameters used by each operations, and the corresponding editable parameters. You can choose a "Preset" formula, such as **RMS** or **Power In Band**, or create your own. By default, the formula is set to **Bypass** (output = input).

The gizmo unary formula consists of up to three stages. If you want to view or customize the stages, select the **Detail** check box. The stages are numbered to reflect the order of operations and you can click or clear the check box for each stage to enable or disable it.



Unary Options Area - Detail View

As stages are enabled or modified the parameter table is updated to show the updated parameters in use. See Using Bounded Parameters for more information on using the table.

When you are happy with a modified formula you can use the **Copy to** button to save it for later use. The name you choose will then appear in the Preset list. This also ensures the formula is included with all files, if you ever wish to export or share the experiment. To show the path where Presets are stored, click the **Path** check box.

Stage 1 and 2

Each of these stages provide a filter and two operations.

Filter Type	Description
Biquad	An IIR filter of a selectable shape (Lowpass, Highpass, Bandpass, Notch) and order. The **Fc-{N}** parameter determines the center frequency. If shape is **Bandpass** or **Notch**, the **Bw-{N}** parameter determines the bandwidth in in fractional bandwidth **Bw-{N}*Fc-{N}**. If shape is **User**, the filter coefficients are defined in a file generated by the user. This must be a CSV text file format with the coefficients in a single row separated by commas, or with a single coefficient in each row.
Parametric	A second order linear filter used to pass or suppress a particular frequency band. The **Fc-{N}** parameter determines the center frequency. The **Bw-{N}** parameter determines the bandwidth in fractional bandwidth **Bw-{N}*Fc-{N}**.
FIR	A finite impulse response filter defined by the coefficient file generated by the user. This must be a CSV text file format with the coefficients in a single row separated by commas, or with a single coefficient in each row.
Smooth	A simple exponential smoothing filter applied to the input. The **Fc-{N}** parameter determines the effective low pass corner frequency.

Operation	Description
Scale/shift	Multiplies by a scalar (**OpParA-{N}** parameter) and adds a shift value (**OpParB-{N}** parameter).
Bound	Output is bound to min (**OpParB-{N}** parameter) and max (**OpParA-{N}** parameter) values.
Square	Output is the square of the input.
Sqr Root	Output is the square root of the input.
AbsVal	Computes the absolute value of the signal.
Sign	Determines the sign of the input and outputs either -1 (signal with negative value), 0 (signal with no value), or 1 (signal with positive value).

Stage 3

This stage provides an exponential smoothing filter and a way to change the data type. The **Fc-3** parameter determines the effective low pass corner frequency of the filter.

Integer

Scales the input by the **OpParA-3** parameter and then converts to an integer.

Logic

The operations available use natural language labels and apply a truth test using editable values in the parameter table for comparison. The operation outputs a "1" if true or a "0" if false.

Select **Logic Out to Alternate** if you wish to have both the logic output and the signal before the logic test available as outputs to the gizmo. For example, if making an RMS threshold detector, you can output both the threshold crossings and the RMS value used for the detection for visualization with other gizmos. In this case, the **Main** output will be the RMS signal and the **AltOut** output will be the logic signal.

Working with Single and Multi-Channel Signals

The gizmo can be used with single or multichannel signals and automatically detects the number of channels in the input signal. The formula is applied independently to each channel. The Unary Signal Processing gizmo can handle up to 96 channels. If more channels are required, use a second gizmo and a Signal Merger gizmo to combine the results back into one processing stream.

> **Tip**
>
> To pick one channel for processing from a multichannel signal, use the Selector gizmo

User Input

Common Use Cases

Create dynamic stores and logic outputs based on inputs from digital I/O bits or a software button. Use this gizmo to store I/O inputs with values defined by user or another gizmo and create fixed-duration, toggled, or edge logic outputs based.

Data Stored	
Epoch events (optional)	Store each input as onset/offset epoch events
Outputs	
StrobeOut	Single channel logic, conditioned output signal
Exclusive (optional)	Single channel integer, button number that was pressed

Gizmo Help Slides

Quick Help Slide 1

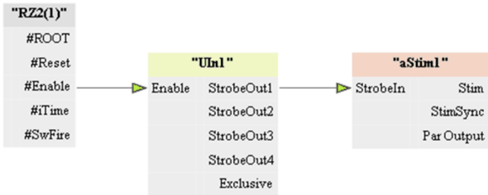
User Input — Create dynamic stores and logic outputs based on inputs from digital I/O bits or software buttons

Clean up incoming digital inputs

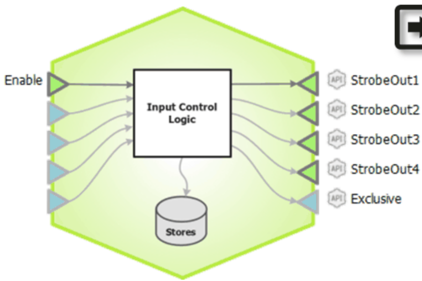
Monitor lever presses and nose pokes

Behavioral box experiments

User Input through button press presents an auditory stimulation



Enable: Activate gizmo. Tie to #Enable on your processor to always stay active



StrobeOut:* Fixed duration, toggled, or edge-detected output for each button press

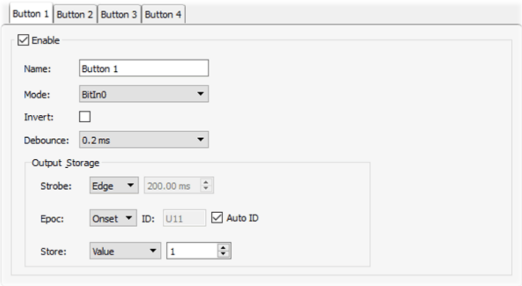
Exclusive: Integer value tells you which button was pressed (1 to 4)

Quick Help Slide 2

User Input

Gizmo Configuration, Data Stored

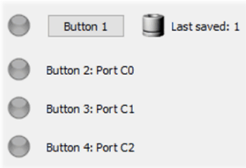
Manual software button toggling and LED indicators for active inputs. Define button parameters and output storage options.



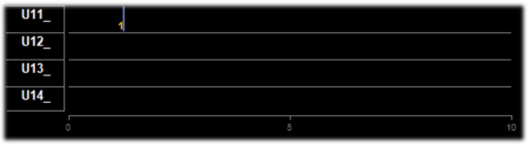
Mode — Choose between bit inputs on the processor or a software button

Debounce — Set a duration of time for an input to settle before being detected (good for lever presses)

Strobe — Control the modes of the StrobeOut (Edge, Fixed, Toggle)



*U1** — Store a gizmo input value, user-defined value, or counter value at the onset or full duration of the button press {Epoch}



Reference

The User Input reads digital inputs from the hardware directly or accepts user button presses and outputs logic signals. The output Strobes can only be true when the Enable input is true. Typically the Enable input is connected to the #Enable signal on your RZ or RX, which remains high for the duration of the recording.

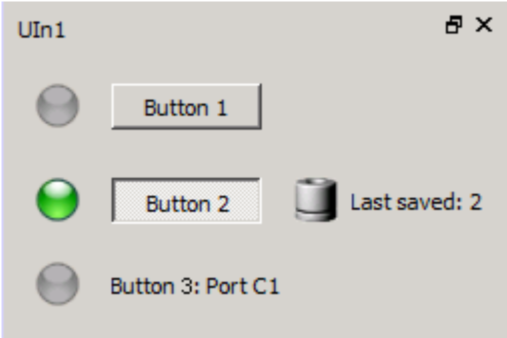
The Runtime Interface

Runtime Plot

At runtime, the standard Synapse data plot is available to display any stored data. The User Input can store its logic signals as onset/offset epoch events.

User Input Tab

Software buttons appear in the user interface at run time. Other bit inputs are labeled with their port name and a state indicator. For inputs with an associated data store, the last value saved is shown on the interface.

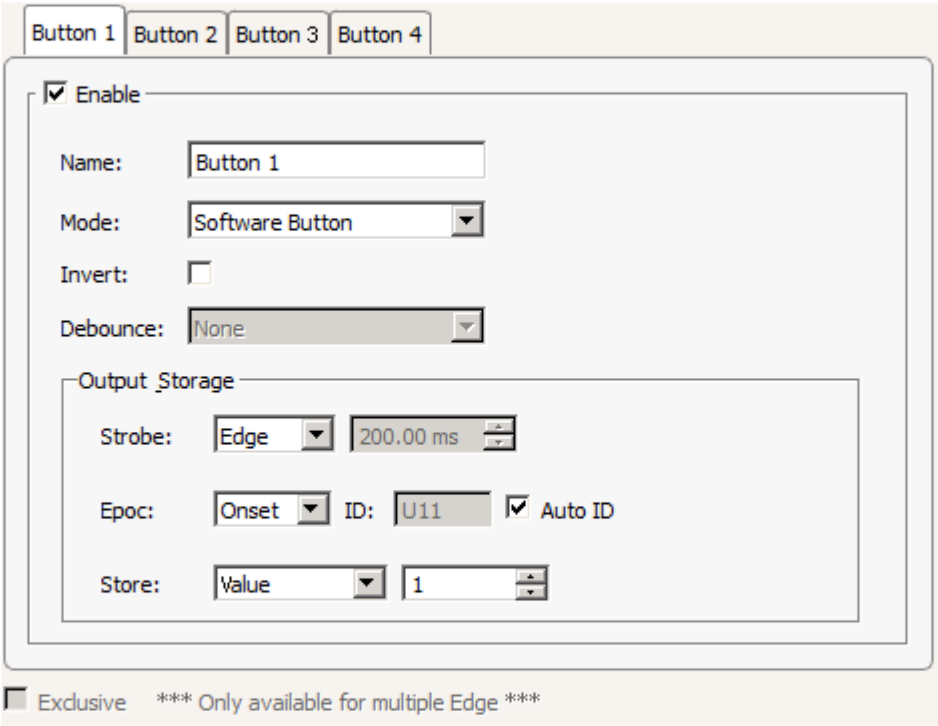


User Input Runtime UI

User Input Configuration Options

See The Options Area and Templates for more information on the gizmo name, source, global options, and displaying the block diagram.

Button Tabs



Button Tab

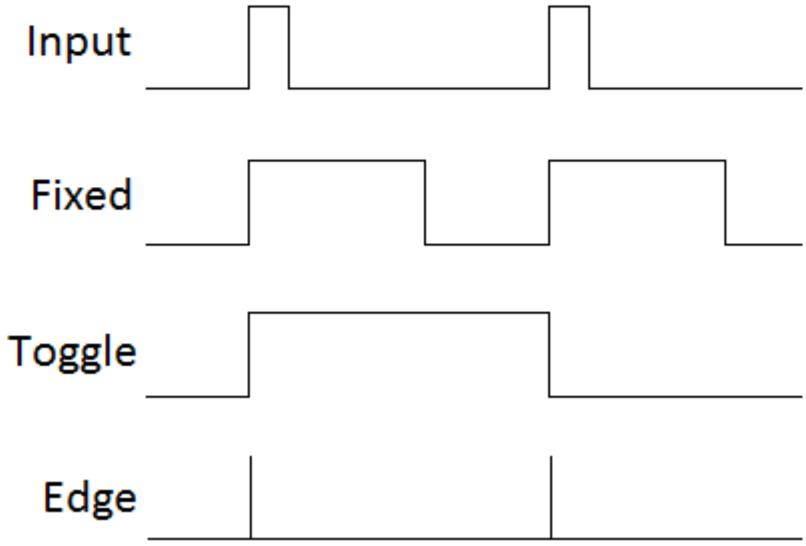
The **Name** is how the input is shown on the user interface.

Button **Mode** sets the input. It can be directly fed from a digital input on the hardware, or set as a software button the user presses.

If **Mode** is a digital input, **Debounce** lets you set a duration of time for an input to settle before being detected (good for lever presses or hardware button presses).

Output Storage

Strobe controls how the button output behaves. An example input signal and the three Strobe options are shown below. Epoc events can be stored on the **Onset** of the Strobe output or on the onset and offset (**Full**).



Exclusive

If all of the buttons are set to **Edge** mode, the Exclusive option is available. When this option is selected, an additional gizmo output is available that contains the value of the button that was pressed (1, 2, 3, 4).

Synapse

Getting Started #1: Setting Up the Rig

Getting Started #1: Setting Up the Rig

Launch Synapse and establish the hardware to software connection. Synapse auto-detects your TDT hardware and uses that information to streamline your experiment design. As you build your experiment, you won't have to think about the particulars of your hardware setup.

Getting Started #2: Edit Screen Overview

Getting Started #2: Edit Screen Overview

A guided tour of Synapse's user interface and an introduction to fundamental processes, like operational modes and user and subject tracking.

Getting Started #3: The Big Gestalt - Part 1

Getting Started #3: The Big Gestalt - Part 1

Get the big picture of how Synapse works and learn about the Processing Tree for experiment design.

Adjust experiment-specific hardware choices, such as the number of recording channels, operational modes, and input sources. As you make selections, Synapse updates the design-time window with relevant choices to complete your configuration.

Getting Started #4: The Big Gestalt - Part 2

Getting Started #4: The Big Gestalt - Part 2

Quickly and easily create experiments with Synapse software.

Getting Started #5: The Big Gestalt - Part 3

Getting Started #5: The Big Gestalt - Part 3

Complete implementation of our demo experiment and preview the run time interface.

Synapse provides function-specific gizmos and experiment templates to automate all but the highest level design steps. For maximum flexibility, build your own custom gizmos.

Getting Started #6: Building Your First Experiment

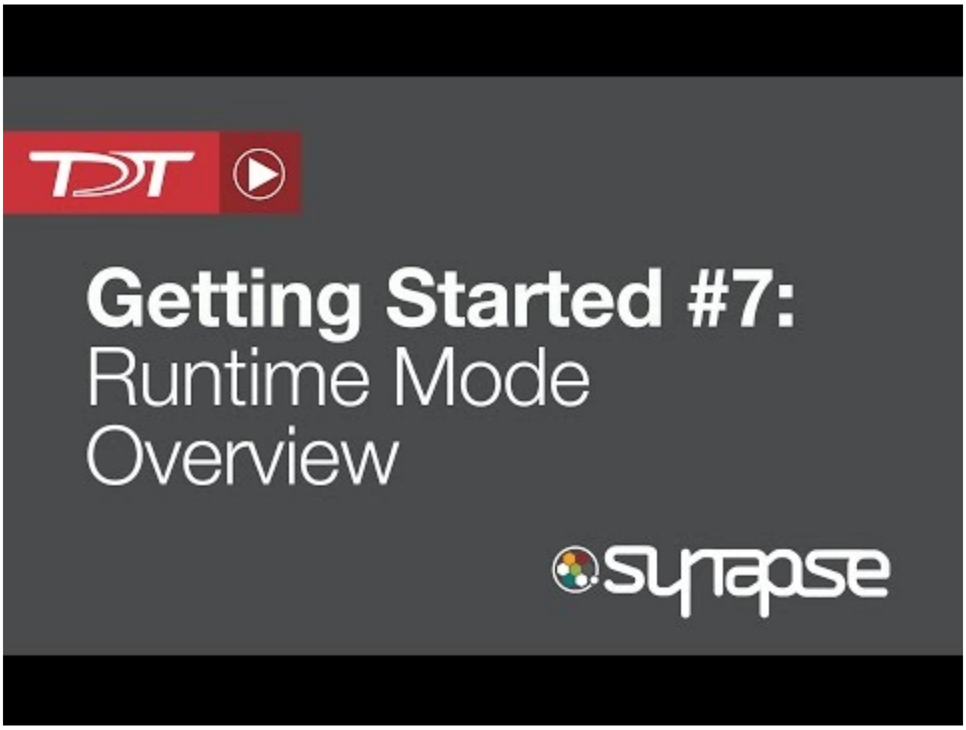
Getting Started #6: Building Your First Experiment

Learn to build an experiment by selecting, adding, and configuring gizmos. This video covers experiment specific hardware settings and different methods of ordering gizmos in the Processing Tree.

Getting Started #7: Synapse Runtime

Getting Started #7: Synapse Runtime

Find out more about Synapse runtime mode, persistence, and user layouts.



How To: Signals & Parameters - Part 1

How To: Signals & Parameters - Part 1

Introduction to using signals and parameters in Synapse.

How To: Signals & Parameters - Part 2

How To: Signals & Parameters - Part 2

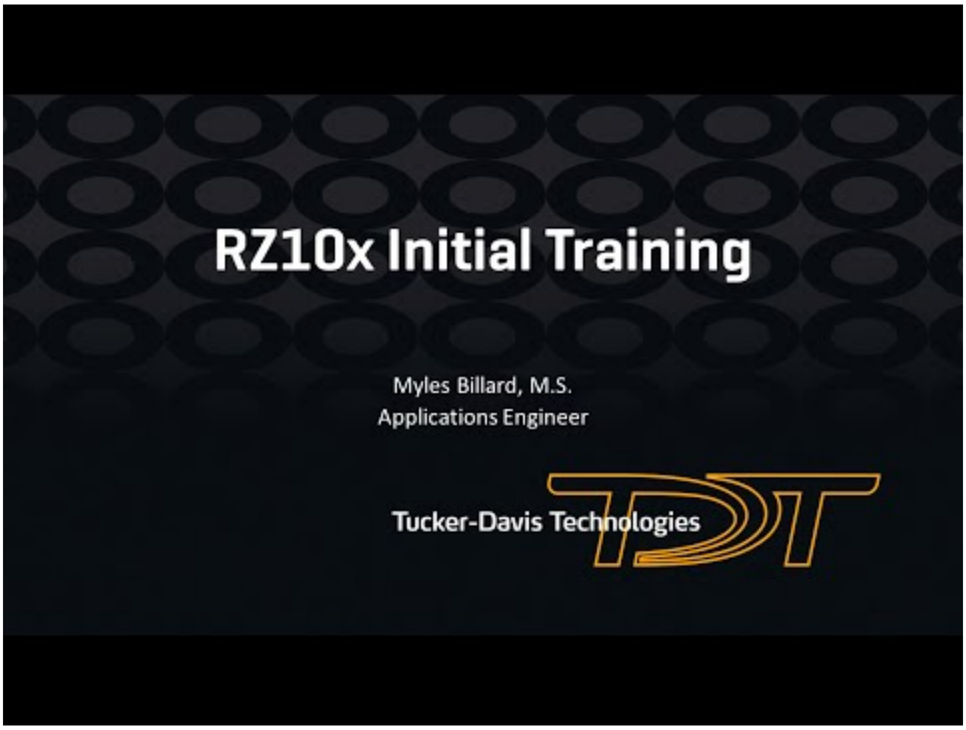
Synapse parameter control including an introduction to the Parameter Sequencer Gizmo.

Fiber Photometry

Fiber Photometry Intro: RZ10x Processor

Fiber Photometry Intro: RZ10x Processor

Get a complete walkthrough of your first experiment using the Synapse Suite Fiber Photometry gizmo and TDT recording system.



Lab Rat and Synapse Lite

Getting Started with Lab Rat - Part 1

Getting Started with Lab Rat - Part 1

Quick tutorial to get you started using the Lab Rat - TDT's all-in-one, portable neurophysiology/ electrophysiology system.

Getting Started with Lab Rat - Part 2

Getting Started with Lab Rat - Part 2

Lab Rat complete setup of your first experiment using Synapse Lite - stimulate and acquire.

Pynapse & iCon

Intro to iCon and Pynapse: Part 1

Intro to iCon & Pynapse: Part 1

Learn how to set up the iCon in Synapse and use the Pynapse gizmo for behavioral experiments. For more information on Pynapse, see the Pynapse User Guide.

Intro to iCon and Pynapse: Part 2

Intro to iCon & Pynapse: Part 2

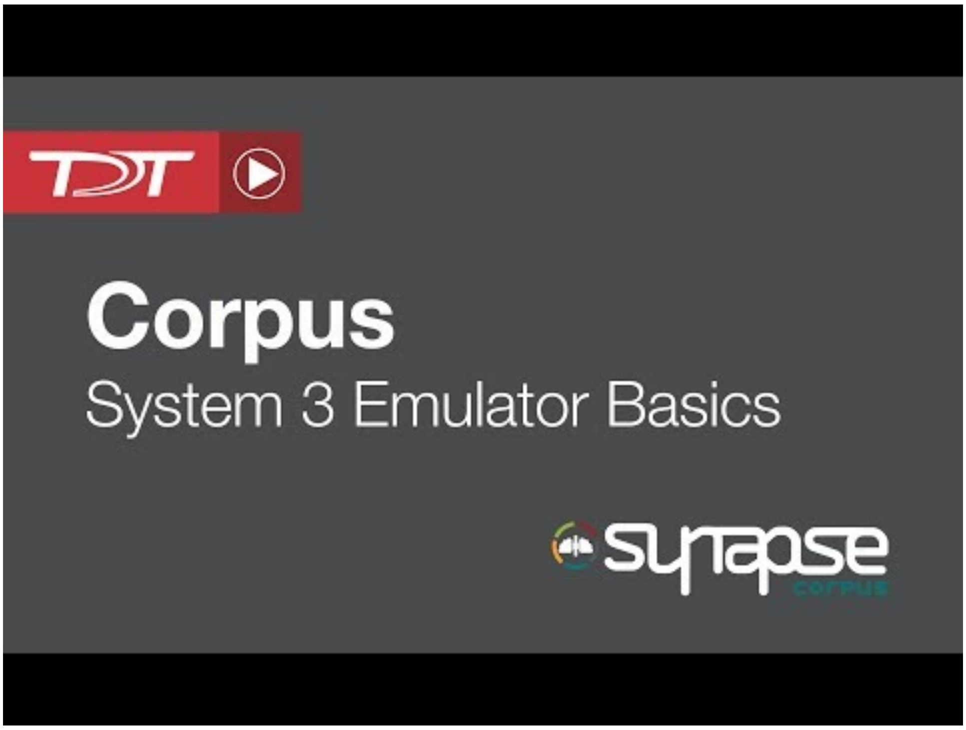
Learn how to use the Pynapse State Machine to create scalable behavioral experiments. For more information on Pynapse, see the Pynapse User Guide.

Miscellaneous

Introduction to Corpus TDT Hardware Emulator

Introduction to Corpus TDT Hardware Emulator

Learn how to emulate TDT System 3 hardware on your PC using Corpus. For more information on Corpus, see the Corpus User Guide.



Cluster Processing with Synapse

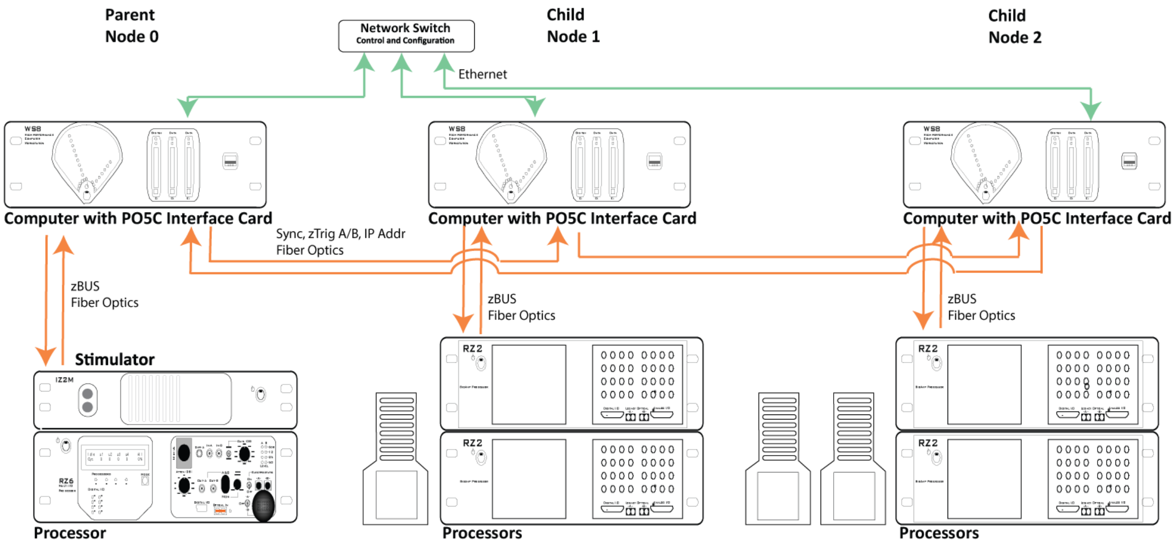
Overview

Cluster Processing is Synapse's solution for research that demands higher channel-count recordings or more processing power. Cluster Processing:

- Eliminates data bottlenecks
- Delivers more PC processing power for data visualization
- Overcomes zBUS bandwidth limitations
- Enables separation of stimulation and recording components, while remaining synchronized

Multiple complete systems, including System 3 hardware and computer, are networked and synched to a single system clock from which acquisition is triggered. This allows a single experiment to be distributed across multiple systems with one system serving as a "parent" node and the other systems as "child" nodes. This is accomplished using PO5c interface cards installed in each system PC. A switch on the card sets the parent/child role for the PC and its related hardware.

The parent node is always node 0. Synapse must be installed and running on all nodes. The PO5c includes a standard zBUS fiber Optic port and a second port that connects the cluster PCs in a "daisy-chain" fashion, delivering clock and trigger signals as well as system IP Address. Once in Cluster mode, The Rig Editor and Processing Tree on the parent computer become tabbed objects. Each tab configures a different node. The operational mode, such as Record or Preview, is controlled on parent node 0 during an experiment.



Cluster Processing Overview Diagram

The network connection is used to send experiments to the child nodes and report status to the parent node.

Setting Up Cluster Computing

Installation

Synapse must be installed on each computer that will be used as a node in the cluster.

To launch in Cluster mode, add "/cluster" to the command line path. For example:

```
C:\TDT\Synapse\Synapse.exe /cluster
```

PC Requirements

All PCs must have an Ethernet, network connection and a PO5c System 3 interface card installed.

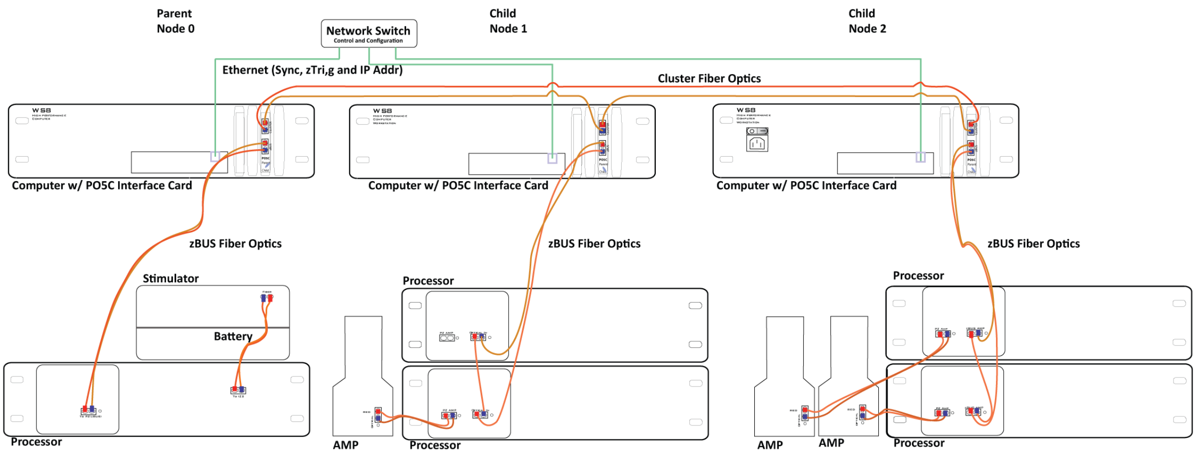
TDT Hardware Requirements

Each node in the system must be a complete system with computer, interface card, and connected RZ processor.

Hardware Configuration

When configuring your hardware, begin by connecting each individual node as you normally would for a single system. See the System 3 Installation Guide for detailed instructions. When the connections within the node are complete, you will connect the nodes together using the CLUSTER fiber optic ports on the PO5c cards as shown in the diagram below.

Connect Node 0 output to Node 1 input and Node 1 output to Node 2 input and so forth and then connect the final node output back to Node 0 input. After all fiber optic connections are complete, connect each node's Ethernet port to a shared network.



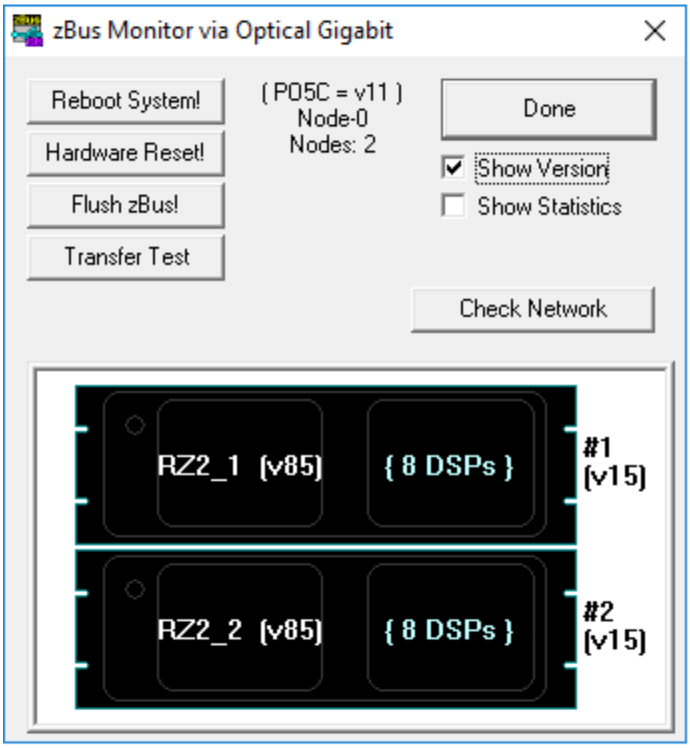
Cluster Computing Hardware Connection Diagram

Each node's role is determined by the parent/child switch on the PO5c card. The switch settings must match the node's position in the chain with Node 0 being the only Parent.

You can confirm that the cluster is correctly configured using the zBUSmon utility.

To launch the utility:

1. Click the zBUSmon shortcut on the parent desktop.
2. Repeat for each child computer.

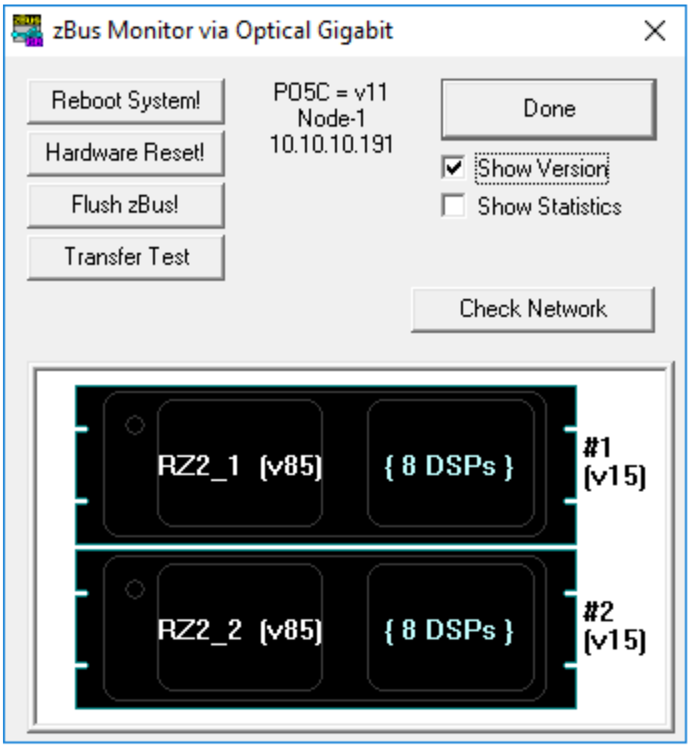


zBUSmon on Parent Node

Parent Node Interface Display

At the top of zBusMon, find:

- Interface Card and Version
- Node number
- Number of nodes



zBUSmon on Child Node

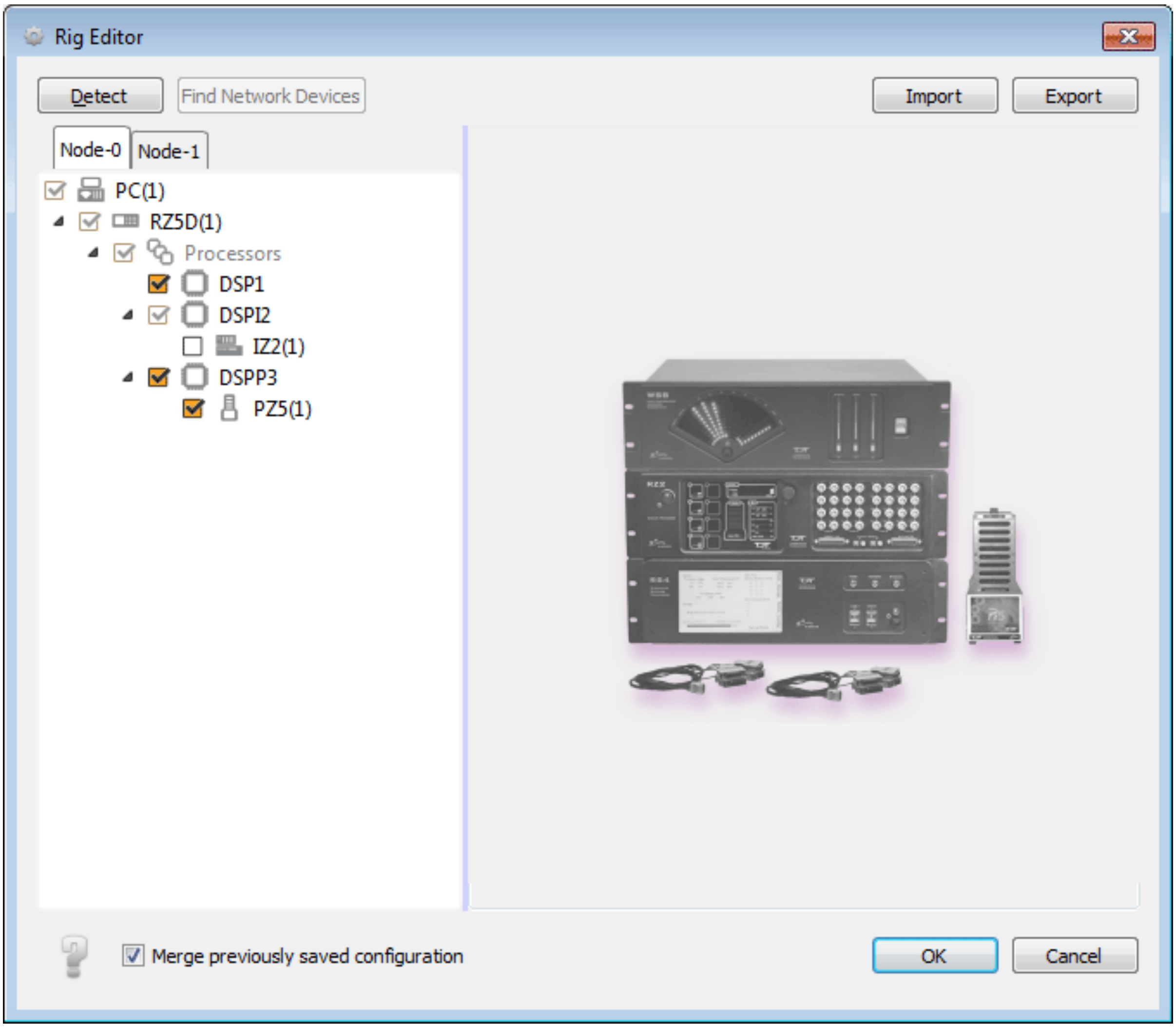
Child Node Interface Display

At the top of zBusMon, find:

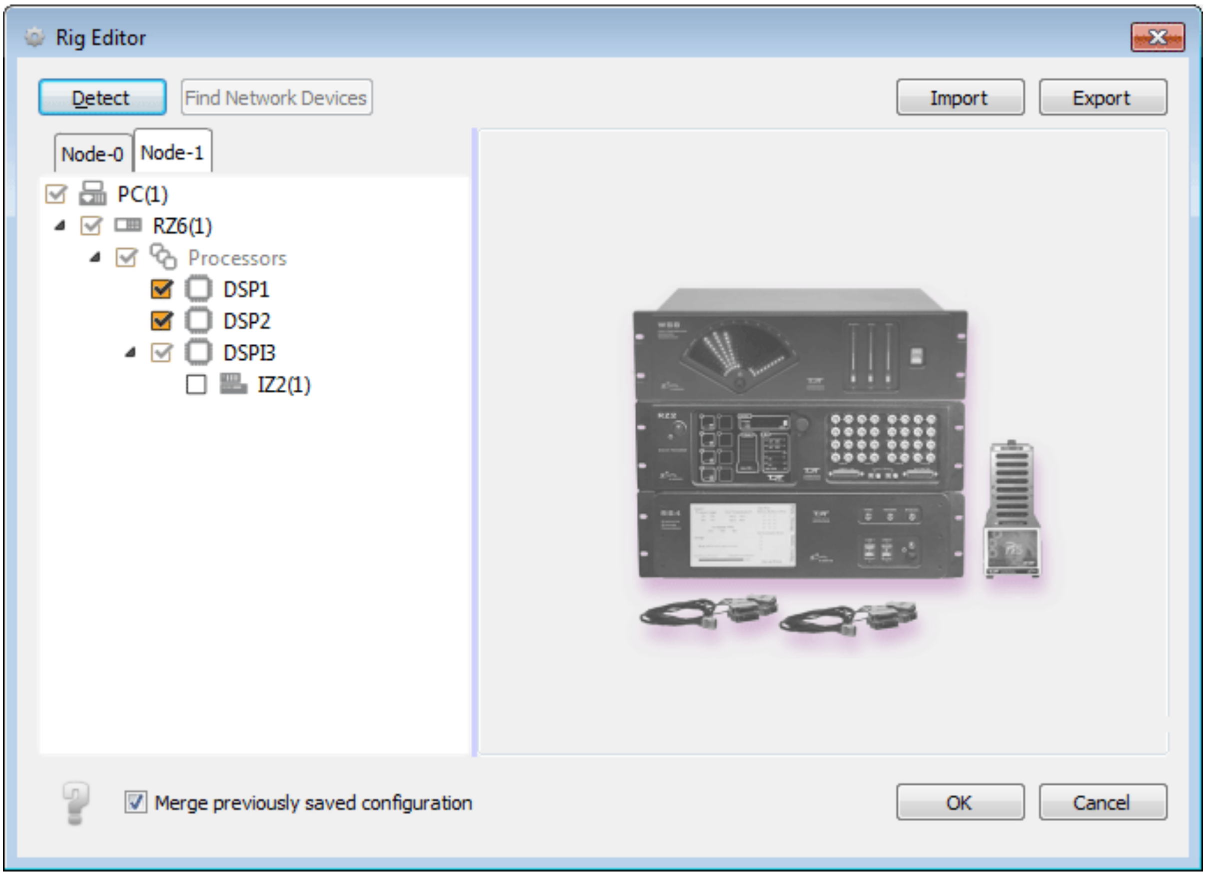
- Interface Card and Version
- Node number
- IP address of Parent (Synapse running on parent node)

The Rig

The Rig Editor is accessed and used much the same in Cluster Mode. If multiple nodes are connected, a tab will be added to the Rig Tree for each node.



Rig Editor: Cluster Mode, Node 0



Rig Editor: Cluster Mode, Node 1

Processing Tree

Processing Tree in Cluster Mode

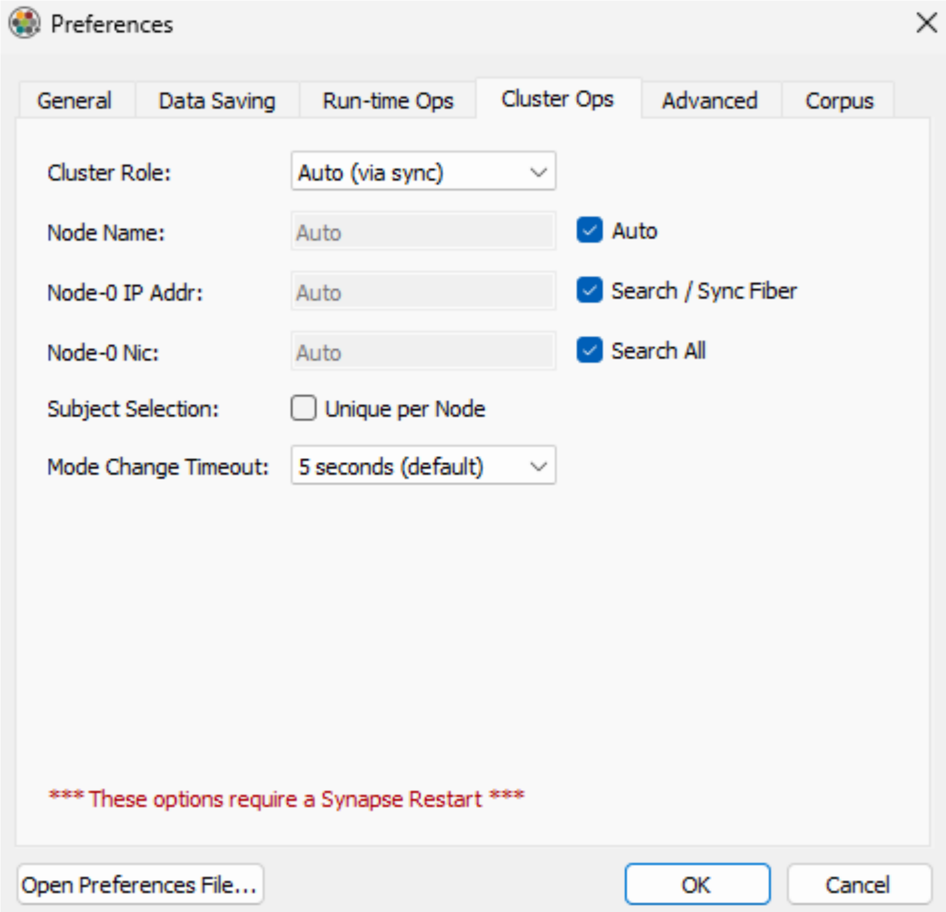
In Cluster mode, the Processing Tree is comprised of a tab for each node - the computer and connected System 3 devices. The experiment is designed using the node 0 computer. Here, you can see the hardware available in each node and use gizmos to create a processing tree for each node.

At the bottom of the Processing Tree pain, icons are displayed for each node. If a node has been previously defined and is not currently connected, the missing asset is shown in gray and the corresponding tab is disabled. The icon also includes indicators for synchronization and connection status via the orange square pulse.

Every Commit compiles not only the parent experiment but also the child experiments. If that is successful, the child experiments are pushed to the child nodes and imported. The import status from all children is returned to the parent.

Preferences

In Cluster Mode a Cluster Ops tab is added to the Preferences dialog box. On this tab, you can chose to manually set the parent/child roles and the name, IP Addr, and Nic for node 0. You can also set the length of the Mode Change Timeout.



Preferences Dialog, Cluster Ops Tab

By default, the user, experiment and subject are common across all nodes. The Subject Selection check box allows you to make the subject different for each node. This is particularly useful when running multiple subjects in a single recording session.

Troubleshooting FAQs

Why do my TTL inputs and/ or data seem delayed by approximately 1 second?

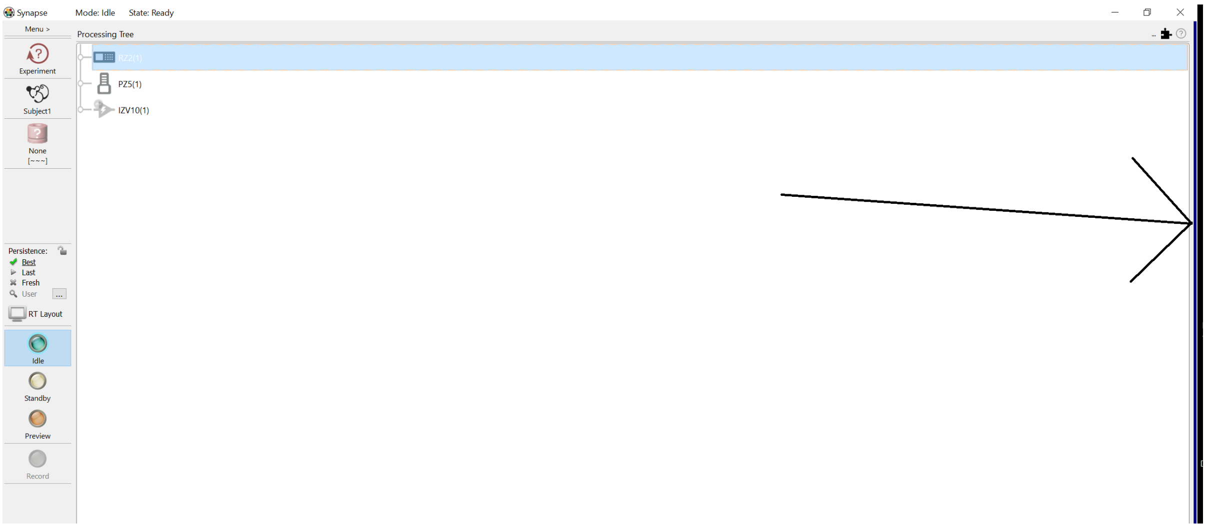
Smooth Scrolling

What you are seeing is a purely visual delay caused by a setting called **Smooth Scrolling**. Smooth scrolling creates an ~1 second delay on inputs in the flow plot (this is purely visual, the data itself is not delayed). This setting is on by default when you install Synapse. You can turn this off in Menu → Preferences → Run-Time Ops → Smooth Scrolling. Here is a link to a Lightning Video showing the effects of smooth scrolling.

I can no longer see any gizmos or gizmo settings

Synapse Window Divider

There is a vertical divider between the Processing Tree and gizmos/ gizmo options in Synapse. This divider is a thin blue line. When the gizmo pane collapses, that line will be on the far right of Synapse window. Click and drag the blue line back to the left to recover the gizmos and gizmo options pane. Clicking on the line can be tricky - the line will become highlighted and darken once you successfully made a click.



Synapse is crashing or erroring out -- how can I fix this?

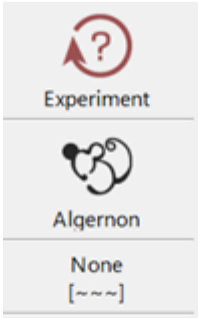
Synapse Crashing or Erroring Out

Please refer to Investigating a Synapse Crash below.

Why is the 'Record' button grayed out in Synapse?

Record Button Grayed Out

The Record button will be grayed out if 'User' (optional), 'Experiment', or 'Subject' information is not selected. These categories will have a red icon in this case. Please select a known User (if applicable), Experiment, and Subject, or save a new one by clicking on the category and picking from one of the options.



A valid experiment name is not selected

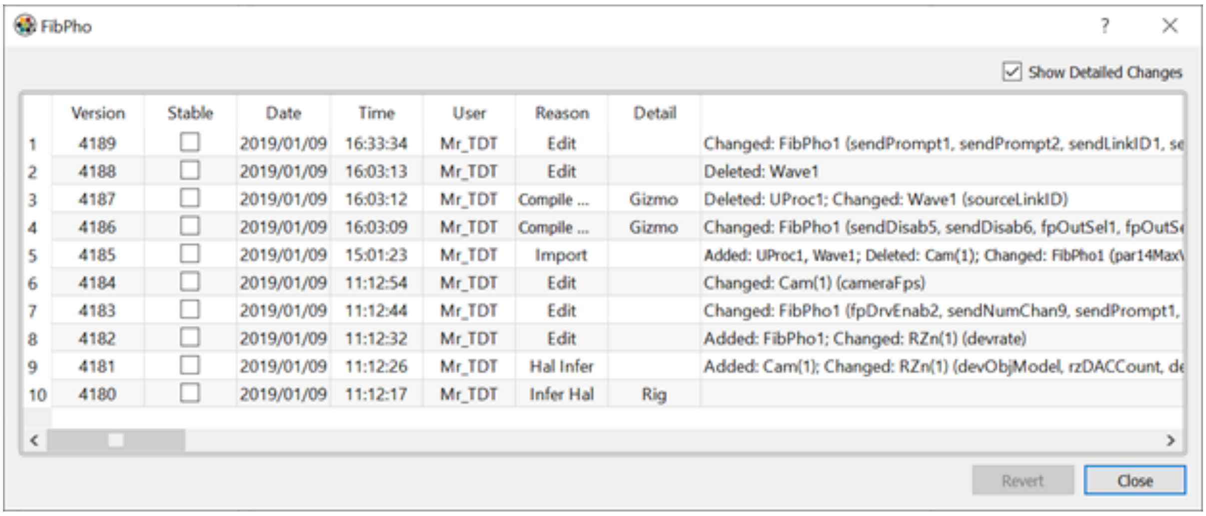
My experiment was working before and now it does not do what I expect – what changed?

Experiment Revision and Persistence

First, make sure you are on the correct experiment name and subject name. Runtime settings are tied to both the experiment and subject name, not just the experiment name. If you are on the correct Experiment and Subject, then you can investigate whether an experiment or its runtime settings have been modified by checking the revision logs or the persistence logs.

Revision Log

The revision log contains the history of all compiled changes made to an experiment. You can access this by clicking on the experiment name and selecting 'Revision Log'



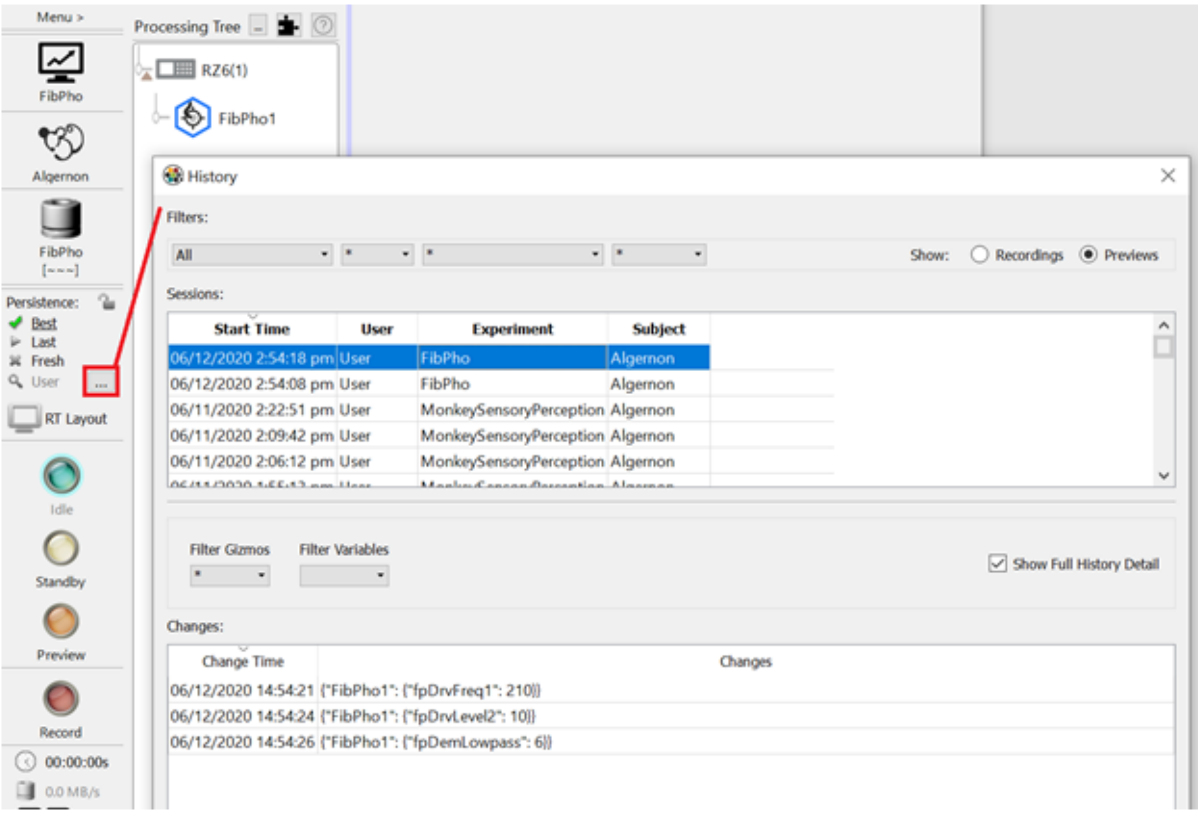
Revision log for the 'FibPho' experiment

If you check 'Show Detailed Changes,' then you can see when gizmos were Added, Deleted, or Changed between each experiment compilation. You can also mark experiment versions as 'Stable,' so that you always have a known good experiment to return to.

Persistence Log

The persistence log ('History') contains the history of all changes made to an experiment during runtime. You can access the persistence by clicking the '...' button on the side bar or in Menu → History. You can filter by 'Record' or 'Preview' mode, and by Date, User, Subject, and Experiment name. For any preview or recording, you can select an experiment and see which settings were adjusted. The readout will have the name of the gizmo and the setting {"Gizmo Name": {"Setting 1": value, "Setting 2": value}}. You can revert to settings used before or after any experiment by right-clicking the experiment in Sessions and choosing 'Use Starting State' or 'Use Ending State.' You can also revert to changes up to a point by right-clicking in Changes and selecting 'Use Changes Up to Here.'

The persistence settings you selected will be enacted the next time you go to Preview or Record mode.



Revision log for the 'FibPho' experiment

We are upgrading our computer to Windows 11. What should I do?

Upgrading to Windows 11

Please refer to the Upgrading Windows for more details.

How can I transfer an experiment onto another computer with Synapse?

Transferring Experiments to Another Computer

First, you must export your experiment by clicking the Experiment name → 'Export'. With the *.synexpz file in hand, you can simply put it onto your other machine and drag + drop it into the Synapse processing tree. See the Import Experiment Lightning Video. Please note that your rig may be different between machines, which can affect how an experiment imports.

All my subjects and experiments are gone! What should I do?

Subject and Experiment Info Missing

First, take a deep breath, then go to C:\TDT\Synapse\Backups. This folder contains, among other things, your last known rig, experiment, and a copy of the synapse database (synapse-1.db). You can simply copy the .db file and put it in C:\TDT\Synapse. Once Synapse is restarted you should have recovered all your experiment and subject information.

You can also find copies of experiments used to record data in the .tin file (located with all the data files for a recording block). The .tin file is really a zip. Change the extension to .zip, then extract the files. You can then import the .synexpz files into your Synapse processing tree. See the Import Experiment Lightning Video.

Will upgrading Synapse ruin any of my experiments?

Upgrading Synapse

No, it will not. All new versions of Synapse (which are free to Synapse license holder) are compatible with older Synapse experiments. There are rare exceptions where a newer version of Synapse will cause a link between gizmos to break because the link has been renamed, but that is easily remedied by contacting support@tdt.com and asking for additional help.

To check for updates, go to Menu → About → Check for Updates. You will first need to install the latest TDT Drivers/ RPvdsEx from the Downloads page on our website and update your TDT processors to the latest microcode. Once that is finished, you can proceed with upgrading Synapse. You can also reference the How to Update Synapse page for more updating details.

Note that MATLAB and all other TDT applications must be closed during the upgrade process.

Why does my gizmo have an orange caution symbol on it?

Deprecated Gizmos

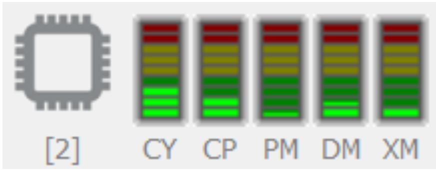
The caution symbol attached to certain gizmos indicates that these gizmos are deprecated. Deprecated gizmos are simply gizmos that have more modern replacements. There is no problem with your experiment if it is running a deprecated gizmo. Current deprecated gizmos include Electrical Stimulation and Local Field Potentials (LFP).

Local Field Potentials gizmo icon showing the deprecated gizmo symbol

I have a '!Warning: Unable to find DSP for this object' message on one of my gizmos. What does this mean?

Unable to Find DSP Warning

DSP stands for digital signal processor. Your TDT processor has DSP cards inside of it that do all the computational heavy lifting. Each gizmo gets assigned to a DSP when Synapse compiles an experiment. On any given DSP there is a maximum number of Cycle Usage (CY), Components (CP) and memory (PM, DM, XM) that can be utilized. Below is a picture showing a gizmo compiled on DSP '2' with the various categories filled out:



Example assignment of a gizmo onto DSP 2. The various DSP category bars are filled based on how much processing, components, and memory are required

When a '!Warning: Unable to find DSP for this object' message appears, this means that Synapse was unable to compile your experiment because it ran out of DSP processing power. In most cases, this is because the Cycle Usage limit was exceeded. There are several ways to try to remedy the situation:

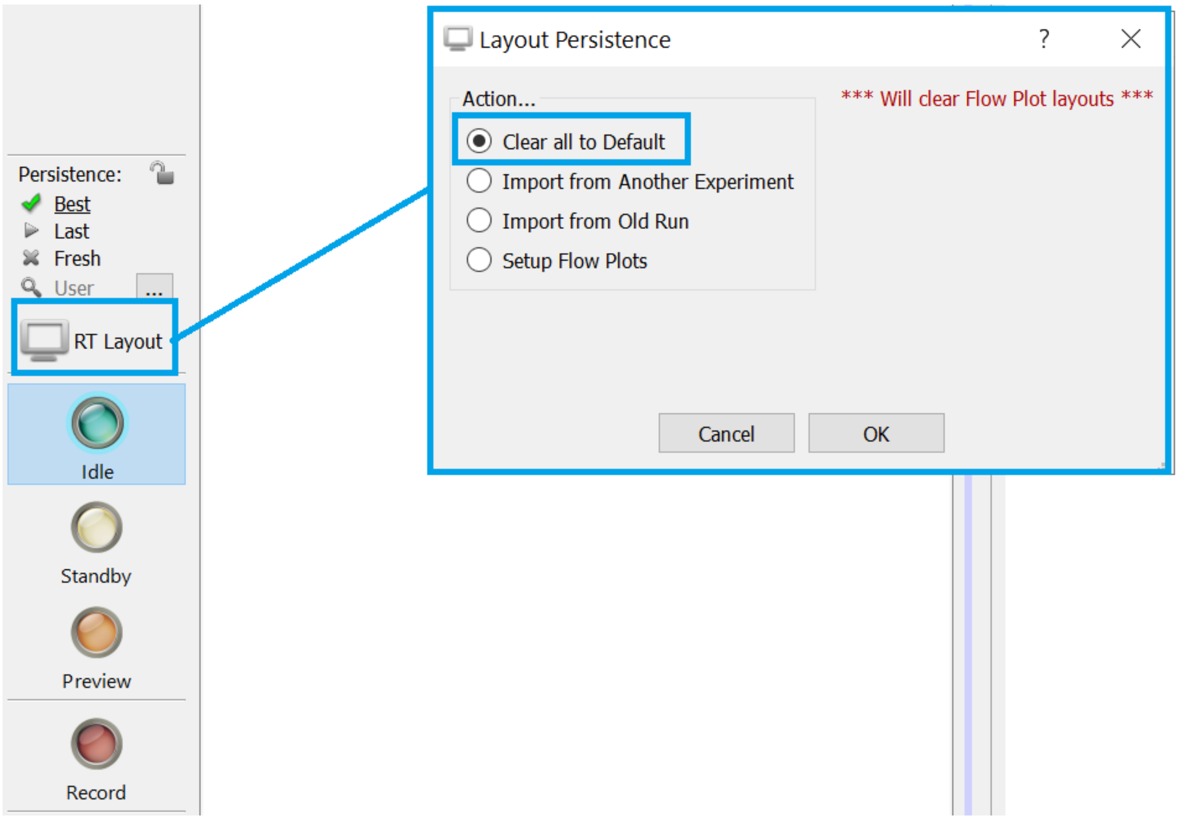
- Check if the Compiler Optimizer is on in Menu → Preferences → Advanced. Also, increase the 'Compiler Passes' parameter from the default 3 to 6 or more.
- Check your RZ or RX processor 'Master Device Rate' in the 'Main' tab. Is the sampling rate for the processor needed for this experiment, or can it be lowered? Often when there is a DSP warning it is because the sampling rate is too high. The cycle usage on a gizmo is directly related to the sampling rate. If CY is 20% full on a gizmo when the sampling rate is 6k, then it will be 40% full when the sampling rate is doubled to 12k.
- If you cannot change the sampling rate, then try unchecking 'Load Optimization' in the 'Main' tab of your RZ gizmo and drag the Processing slider all the way to the right (for v92 of Synapse or higher). This will increase Synapse's CY threshold before it decides it cannot compile an experiment. This does not increase the physical limit on the DSP, however, and in most cases 'Load Optimization' should be on.

If these options do not work, please contact support@tdt.com for more help.

All my flow plots are gone or are blank. What should I do?

Reset Flow Plot

Sometimes your flow plot may be blank and your traces stop showing. First, try autoscaling a trace to make sure it isn't just shifted. If that does not work, then you will need to reset your flow plot. To do this, go to Idle → RT Layout → Clear all to Default. This will reset the flow plot. Your plots should reappear the next time you go to Preview or Record.



Reset your flow plot in Synapse

Investigating a Synapse Crash

First Things to Check

If you are experiencing a Synapse or TDT program crashing or erroring out, here are some quick things to check first which may solve the issue:

1. Check the version number of all TDT programs installed. Browse to Windows Programs and Features to look. All TDT software will have a Tank icon and the publisher will be Tucker-Davis Technologies.

   Mismatches in program version numbers can cause crashes because certain files are shared between programs. If there is a mismatch, please go to our Downloads page update everything to the latest version.

SynapseLite v90	Tucker-Davis Technologies	9/24/20
SynapseSuite v91	Tucker-Davis Technologies	5/16/20
TDT ActiveX Control for PA5 v90	Tucker-Davis Technologies	9/24/20
TDT ActiveX controls v90	Tucker-Davis Technologies	9/24/20
TDT BioSigRP v101	Tucker-Davis Technologies	9/24/20
TDT BioSigRZ v90	Tucker-Davis Technologies	1/25/20
TDT Drivers v91	Tucker-Davis Technologies	4/22/20
TDT OpenBridge v90	Tucker-Davis Technologies	9/24/20
TDT OpenDeveloper v90	Tucker-Davis Technologies	9/24/20
TDT OpenEx Software v90	Tucker-Davis Technologies	9/24/20
TDT OpenExplorer v90	Tucker-Davis Technologies	9/24/20
TDT OpenSorter v90	Tucker-Davis Technologies	9/24/20
TDT SigCalRP v90	Tucker-Davis Technologies	10/12/2
TDT SigGenRP v90	Tucker-Davis Technologies	9/24/20
TDT SpikePac v90	Tucker-Davis Technologies	9/24/20
TDT SykofizX v90	Tucker-Davis Technologies	9/24/20
TDT TTankMin v90	Tucker-Davis Technologies	3/4/201

   You should also check for general TDT updates. In Synapse go to Menu → About → Check for Updates to see if there is a new release of Synapse available. If applicable, also update your TDT Drivers/ RPvdsEx and DSP microcode following the instructions here.

2. Check your computer power settings. Your sleep and monitor settings should be set to 'Never', and your PCI Express → Link State Power Management should be set to 'Off'. Please see Tech Note 0174 for more detail.

Restoring Synapse in the Event of a Complete Computer Crash or Computer OS Upgrade

Synapse information is easily restored or transferred between computers because everything in Synapse (besides the data itself) is saved to a database file (.db). The db file will help to restore all your experiments, subjects, and runtime settings. There is a folder `C:\TDT\Synapse\Backups` that contains:

1. The last hardware rig you used (.synrig)
2. A UserSettings.ini file
3. The Synapse database file (.db).

To restore your Synapse configuration on any computer:

1. Install TDT Drivers and Synapse on the new machine.
2. Replace `synapse-1.db` in `C:\TDT\Synapse` with the .db file from your Backups folder. TDT recommends renaming synapse-1.db to synapse-1.db.old first, and then just dropping in the new `synapse-1.db` from Backups.
3. Import the rig file (.synrig) via Menu → Edit Rig → Import.
4. Last, you can place your older UserSettings.ini file in `C:\Users\{username}\AppData\Local\Tucker-Davis Technologies\Synapse` to restore all your Menu → Preferences settings.

If you are upgrading your Windows OS, please refer to the Upgrading to Windows 10 document for full instructions.

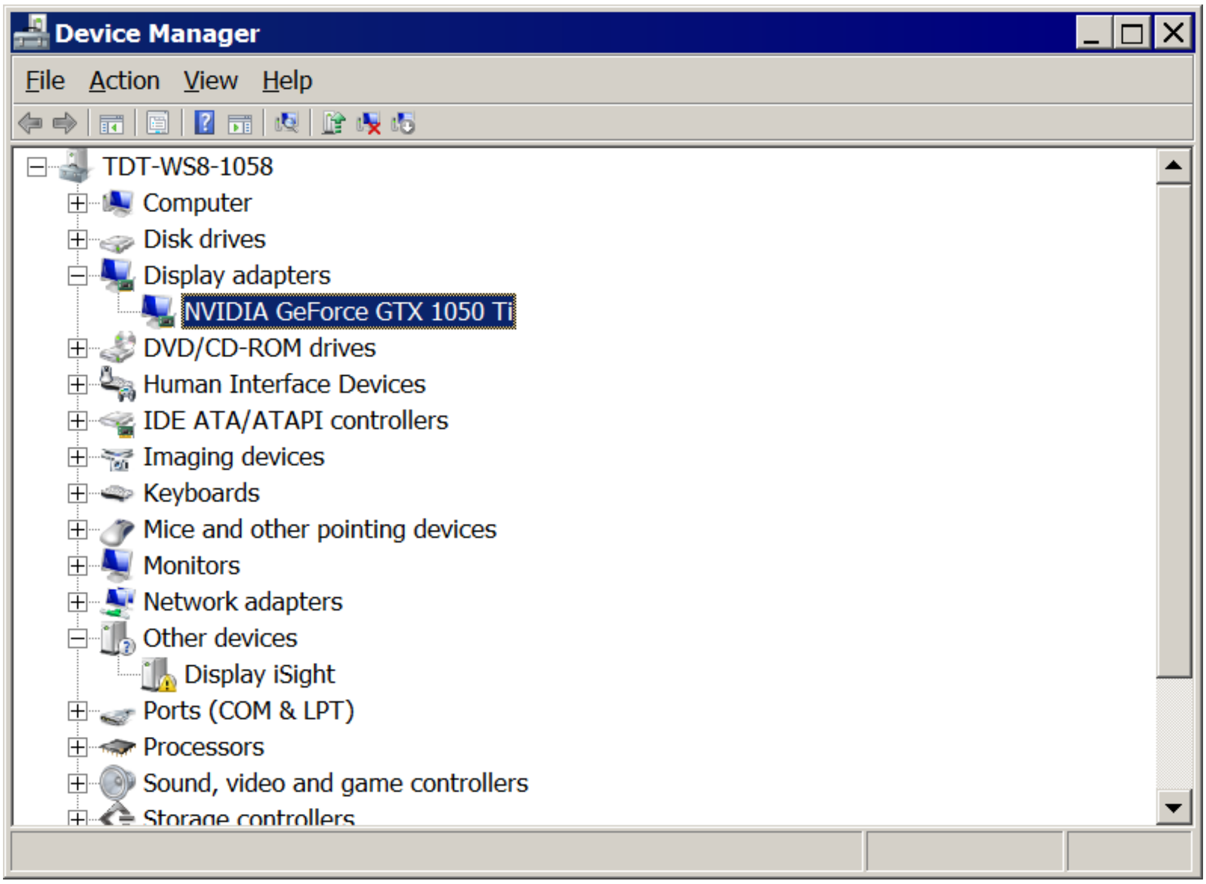
Synapse Crashes When Going from Preview/Record to Idle

Symptom:

Synapse crashes after doing a session in Preview or Record mode when the user tries to go to Idle.

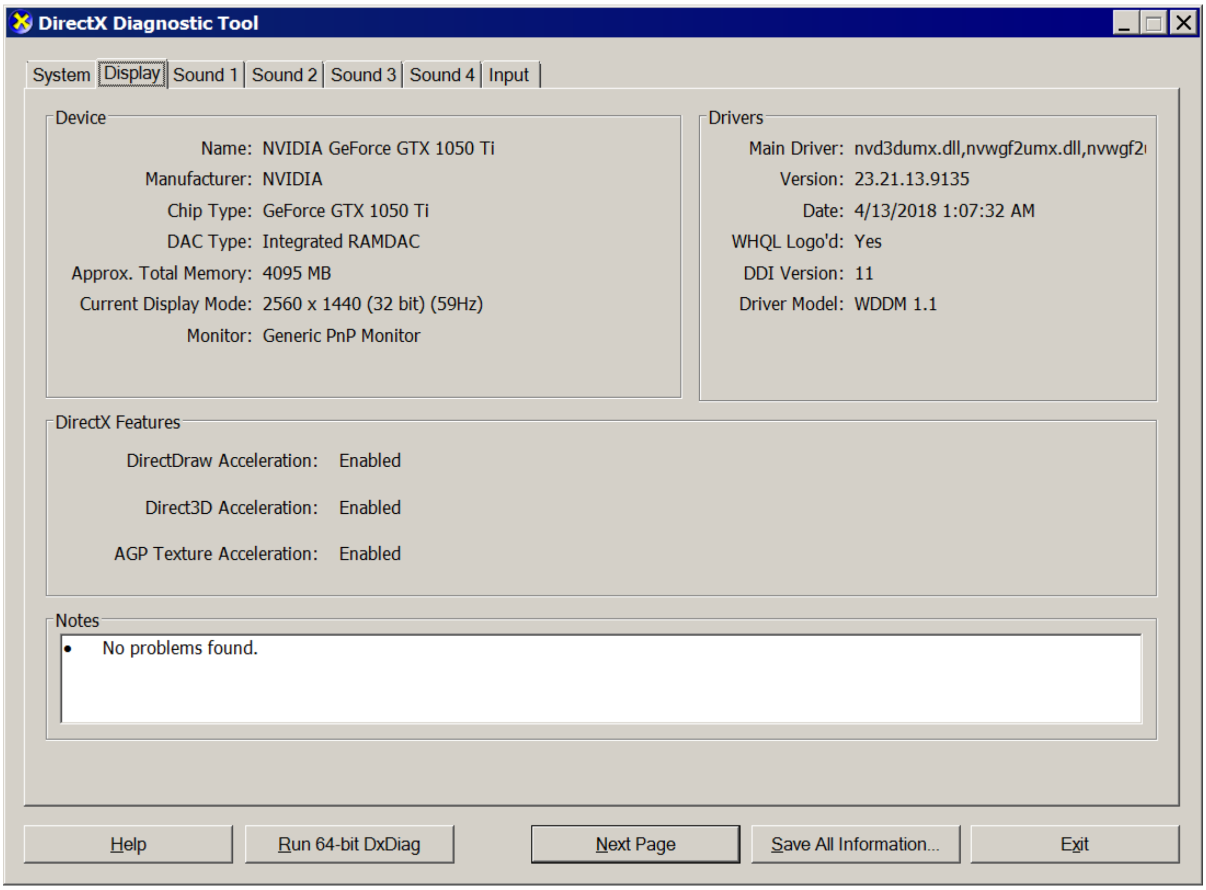
Possible Causes:

The most common problem is the graphics card (GPU). This crash typically happens when using *PCA Spike Sorting* or a *Strobe Store* gizmo with runtime plotting. These gizmos use advanced plotting modules. If you have an older GPU, outdated GPU drivers, or an integrated CPU graphics, then Synapse might crash.



Solutions:

If you have a GPU, then try updating the GPU drivers. Be sure to look up the exact type of GPU and install the correct drivers. Often, if the customer has a NVIDIA or AMD card, there will be programs (GeForce, Radeon) installed to help find new drivers. You can figure out which graphics card is installed by going to Device Manager Display adapters (see first image). You can also run the *dxdiag* command in the windows Start Run dxdiag and check the Display tab (see second image).



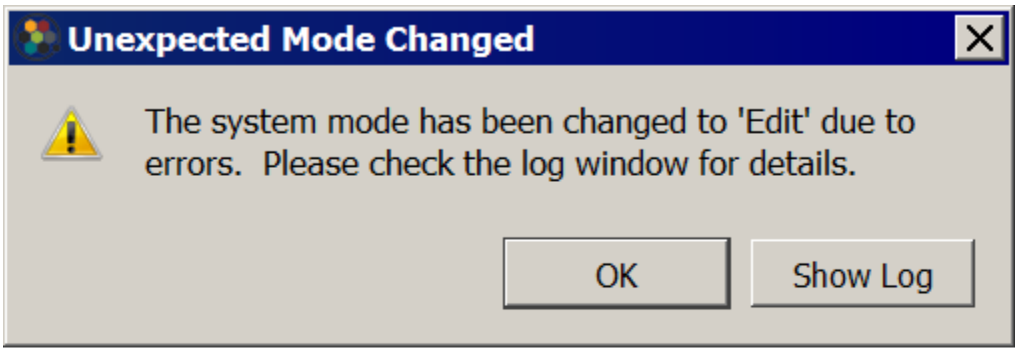
If you have an integrated graphics processor (IGP), then you can also try to update the drivers.

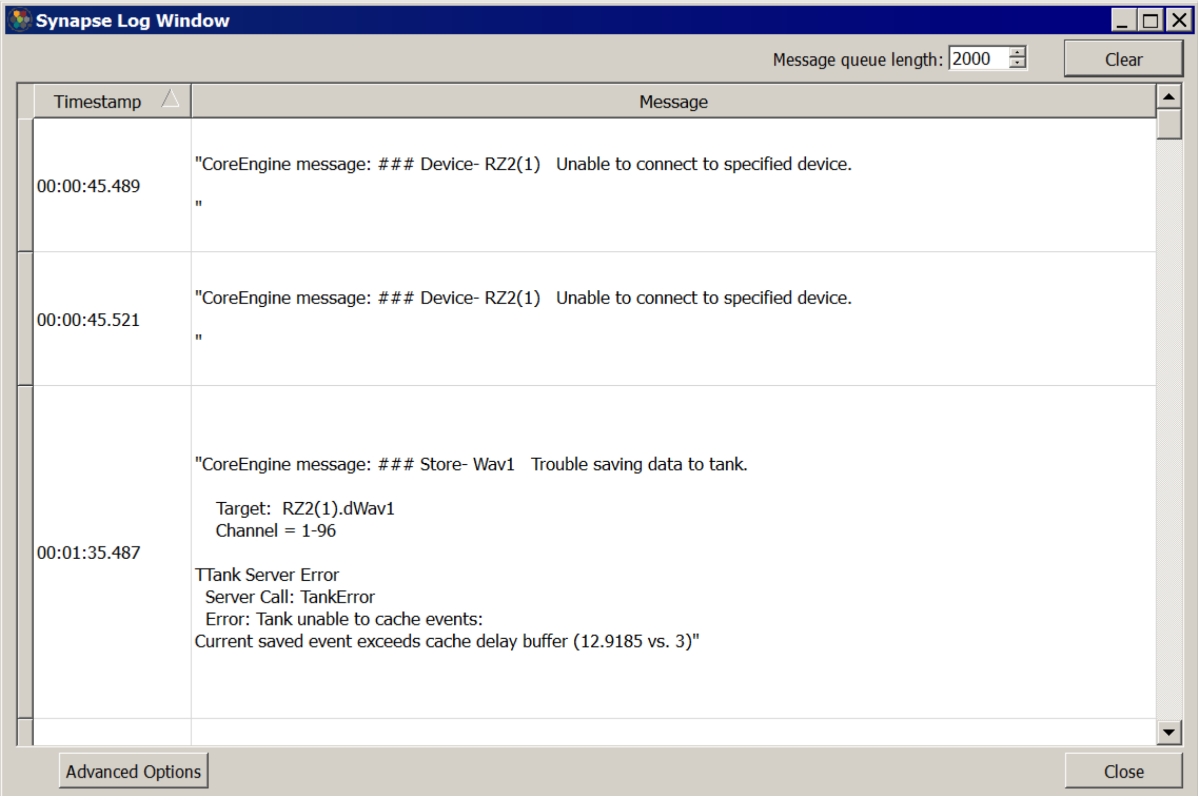
If all else fails, try to run the experiment on a different computer with a dedicated GPU that has up-to-date drivers. If the experiment works without crashing, then the problem is the graphics.

Synapse Errors Out During Preview/Record

Symptom:

Synapse errors out during a recording session (Preview or Record mode). Synapse displays "Trouble saving data to Tank" or "Tank unable to cache events" error message. During runtime there may also data dropouts.





Possible Causes:

Most of the time, this error is caused by trying to save too much data to disc. Whenever the Synapse Tank Server writes to disc it checks the timestamp of the data. If the new data has a timestamp that is above a certain lag (the cache delay) from the last data written to disc, Synapse will report an error that it is falling behind (has limited readback/ trouble saving data to tank). The general guidelines for max data saving to disc are approximately 64 channels total x 24 kHz. If more data needs to be saved, then an RS4 might be needed.

In some cases, however, this error is caused by disconnection of the zBus equipment (RZ, RX) or a bad hard drive write.

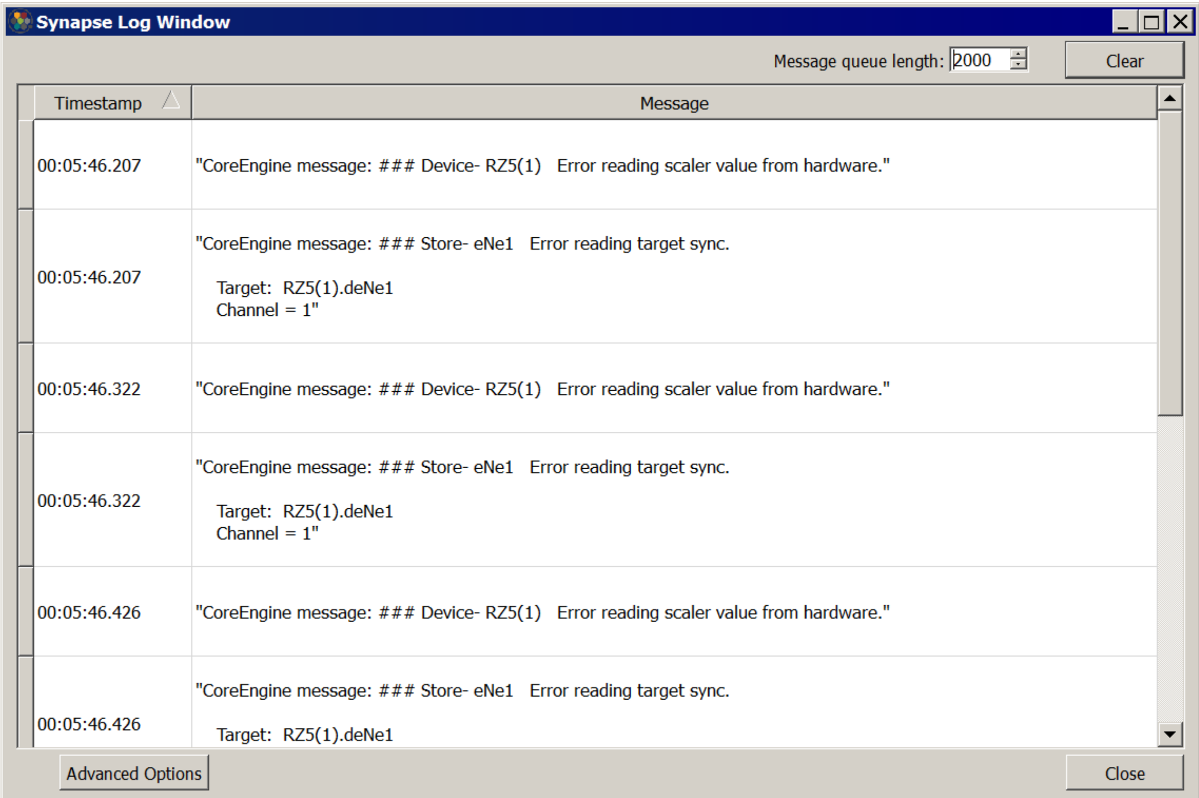
The "Error reading scaler value from hardware" message could be a bad optical connection. Try reseating all fiber optic connections in your system and try again. This error can also occur if the RZ is not getting enough power, which can happen if a power hungry machine is connected to the same electrical outlet.

Solutions:

Tech Notes 0963 goes over this error and possible fixes in detail. Please follow the recommendations depending on the specific type of cache event error.

Synapse Errors Out When Opening Another TDT Program (BioSig, RPvdsEx, zBusMon, OpenEx)

Symptom:



Synapse errors out when another program that tries to communicate with hardware is opened. Such programs include BioSigRZ, BioSigRP, SigGenRZ, SigGenRP, RPvdsEx, zBusMon, OpenEx, and SykoFizX. Synapse will quit recording and errors such as "Error reading scaler value from hardware" and "Error reading target sync" will appear.

Possible Causes:

This happens because the other programs listed are attempting to communicate with the zBus hardware (RZ, RX) at the same time as Synapse.

Solutions:

Do not open any TDT programs that actively try to communicate with the hardware. Other programs, such as OpenScope, OpenSorter, OpenExplorer, OpenBrowser, OpenBridge are okay to open.

OpenScope, OpenExplorer, OpenSorter, OpenBrowser, OpenBridge Cannot Launch or Crash Upon Launch

Symptom:

Peripheral TDT software (Scope, Explorer, Sorter, Browser, Bridge) cannot launch properly or crash upon launch.

Possible Causes:

This is most likely due to a mismatch in software and TDT Driver versions.

Solutions:

Check the version number of all TDT programs installed. Browse to Windows Programs and Features to look. All TDT software will have a Tank icon and the publisher will be Tucker-Davis Technologies.

Mismatches in program version numbers can cause crashes because certain files are shared between programs. If there is a mismatch, please go to our Downloads page update everything to the latest version.

SynapseLite v90	Tucker-Davis Technologies	9/24/20
SynapseSuite v91	Tucker-Davis Technologies	5/16/20
TDT ActiveX Control for PA5 v90	Tucker-Davis Technologies	9/24/20
TDT ActiveX controls v90	Tucker-Davis Technologies	9/24/20
TDT BioSigRP v101	Tucker-Davis Technologies	9/24/20
TDT BioSigRZ v90	Tucker-Davis Technologies	1/25/20
TDT Drivers v91	Tucker-Davis Technologies	4/22/20
TDT OpenBridge v90	Tucker-Davis Technologies	9/24/20
TDT OpenDeveloper v90	Tucker-Davis Technologies	9/24/20
TDT OpenEx Software v90	Tucker-Davis Technologies	9/24/20
TDT OpenExplorer v90	Tucker-Davis Technologies	9/24/20
TDT OpenSorter v90	Tucker-Davis Technologies	9/24/20
TDT SigCalRP v90	Tucker-Davis Technologies	10/12/2
TDT SigGenRP v90	Tucker-Davis Technologies	9/24/20
TDT SpikePac v90	Tucker-Davis Technologies	9/24/20
TDT SykofizX v90	Tucker-Davis Technologies	9/24/20
TDT TTankMin v90	Tucker-Davis Technologies	3/4/201

If your programs are all on the same version and you still cannot launch the program, try running the executable for that respective program in `C:\TDT\Synapse\OpenEx`. If this works, then change the shortcut for that program.

Remote Experiment Design

Overview

This is a guide to using Synapse without connected hardware. This document is designed to help TDT users learn how to setup and use Synapse on another computer for experimental design and testing. TDT provides the Corpus emulation software with the installation of TDT Drivers/ RPvdsEx. Corpus can be used to emulate your existing RZ and PZ hardware, thus allowing users to run gizmos and data through a Synapse experiment as if they were using real hardware.

This guide will take you through installation of Synapse and Corpus on another computer, setting up your exact TDT rig, and run you through an example experiment. If you need assistance with any part of this document, feel free to email support@tdt.com or call into our office (+1.386.462.9622) and a TDT Tech Support Engineer will gladly assist.

Installing TDT Drivers and Synapse on another Computer

Installation instructions will be split into two paths: if you have the USB stick with TDT software or if you do not.

If you have your original TDT USB Stick that came with your system

Then you can simply use the built-in installer on your USB on your new computer. You will want to make sure to be using the same version as your Lab rig's Synapse (check in Menu → About). If your USB version is out of date, you can install the software and then update Synapse by first downloading TDT Drivers/ RPvdsEx from our Downloads webpage, and then from Synapse going to Menu → About → Check for Updates and installing the latest version of Synapse.

If you do not have your original TDT USB Stick

Then you can download everything from online.

First, you must install TDT Drivers/ RPvdsEx from our Downloads webpage.

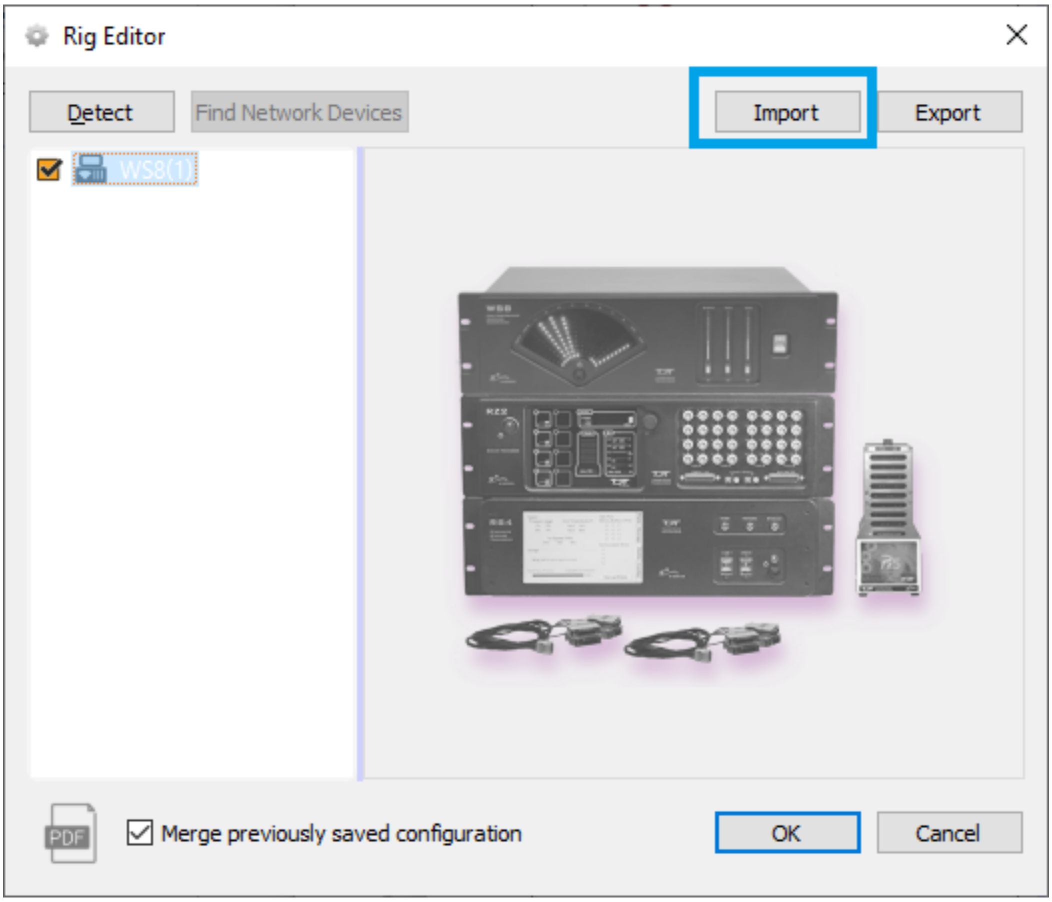
After installing TDT Drivers, you can then proceed to install Synapse Essentials on your machine. Please contact support@tdt.com for the password and have the original PI or purchaser information to verify purchase.

Setting Up Corpus for Your Rig

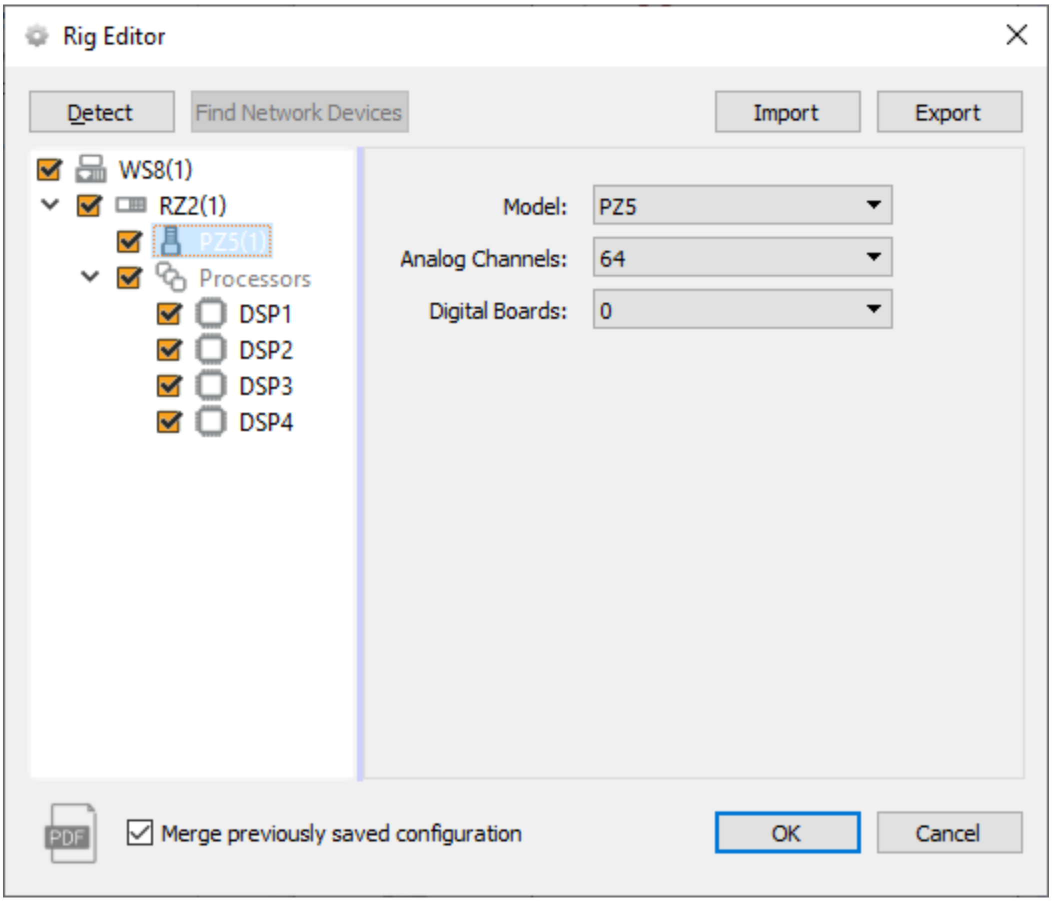
Before you start, you will want to have your existing rig from your dedicated TDT computer. You can retrieve this in Synapse by going to Menu → Edit Rig → Export.

If you have any experiments that you specifically want to work on, you can also export them from Synapse by clicking their name Export (https://www.tdt.com/docs/lightning/synapse/#export-an-experiment)

1. Open Synapse. The rig editor should appear, and it will be blank. In this window, proceed by importing your rig (***.synrig** file) and setting up any peripheral devices (PZ5 in particular) for the correct number of channels.



Open Synapse → Import Rig



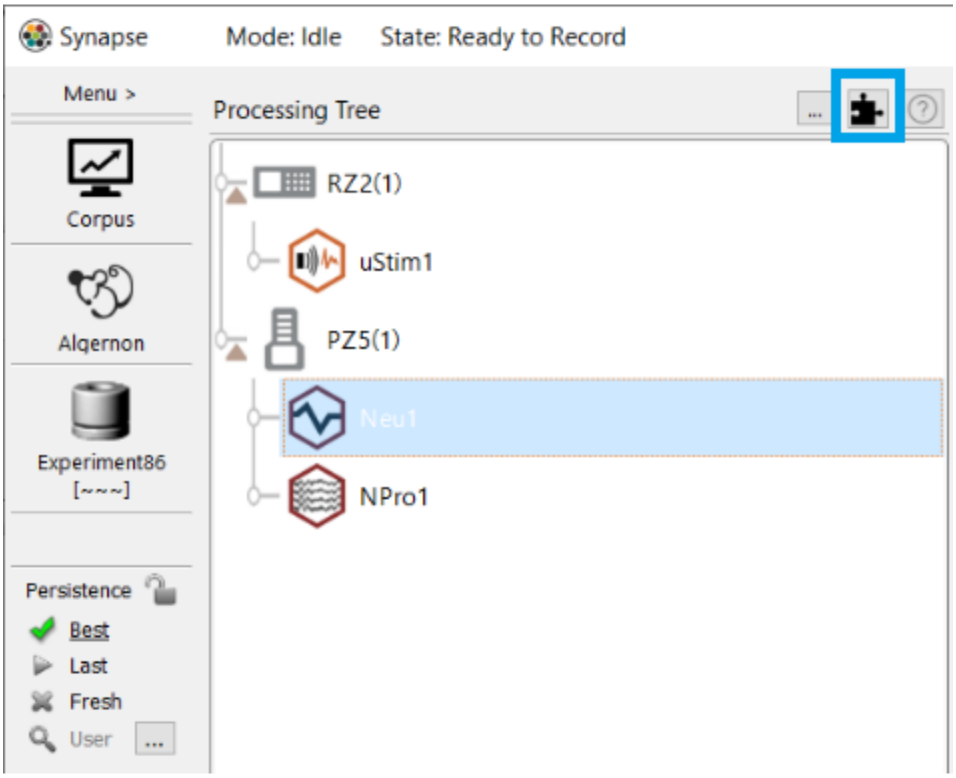
A Setup Rig

If you do not have a Rig file, you can right click WS8 and add RZ devices and DSPs as you see fit.

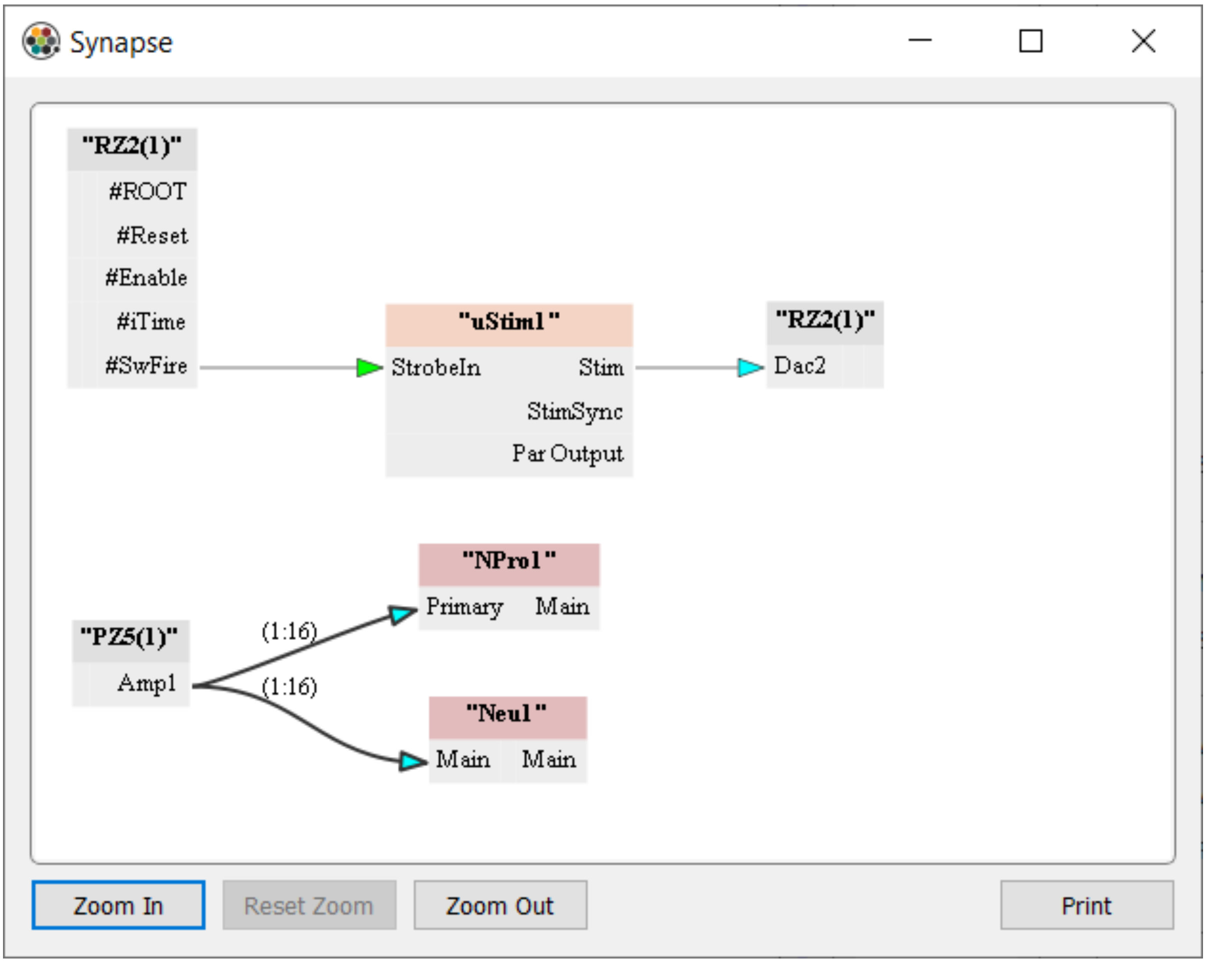
2. Press 'OK' in the rig editor and then launch Corpus. Make sure that the correct devices appear in the Corpus window.

Example Experiment

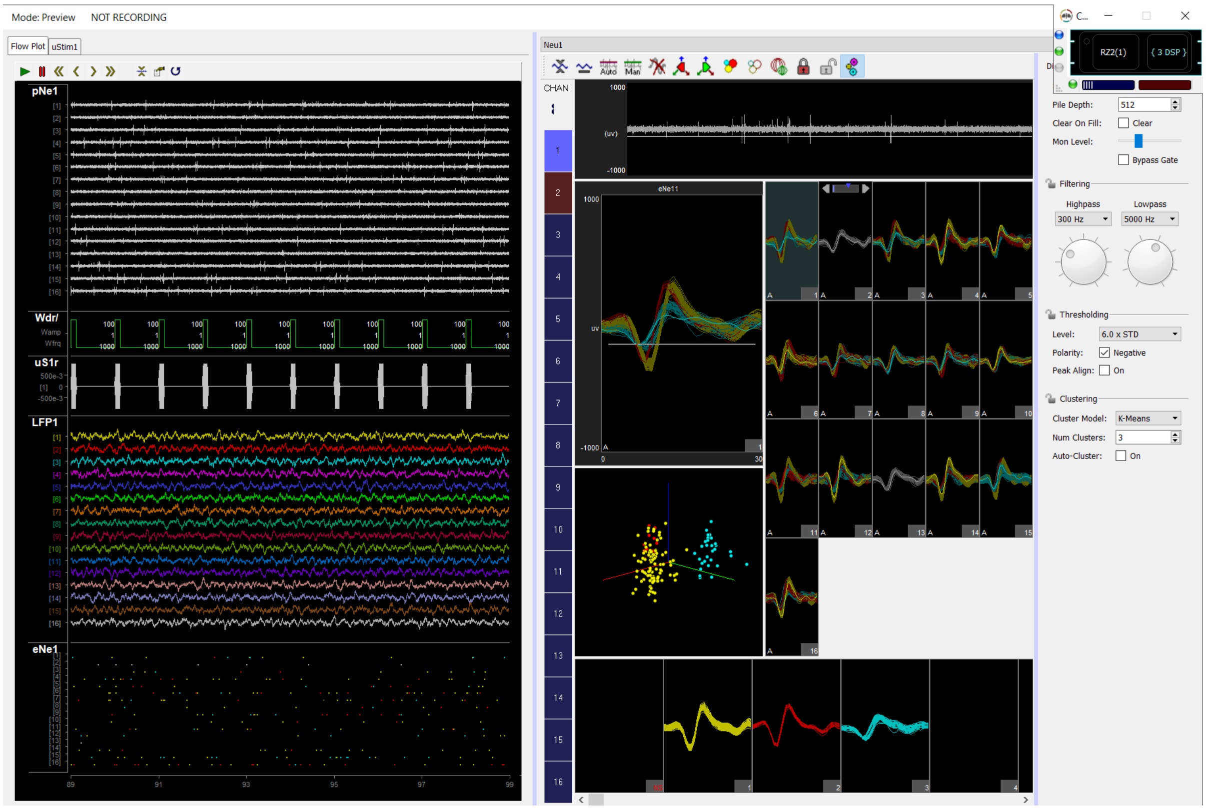
Here is a simple example experiment showing audio stimulation with online spike sorting and LFP filtering. The PZ5 will send fake data out to the PCA Spike Sorting and Neural Stream Processor gizmos. The Ultrasonic Stimulation gizmo (uStim) is routed to the DAC output of the RZ2 in preparation for export to the main lab rig in the future.



Example Processing Tree



Example Connections Diagram



Example experiment running with fake data generated by Corpus

Using Your Own Data

You can add your own data in Corpus to replay previously recorded ADC or PZ5 signals. Please see the 'Inputting Fake Data' section of the Corpus User Guide for more details.

If you have MATLAB on your computer, you can also use the SynapseAPI and this example script to export continuous data to Corpus. Please be sure to have the MATLAB SDK installed on your computer and make sure it is in your MATLAB path.

Exporting Your Experiment to Another Computer

Exporting your new experiment is simple. Just click the experiment name → export. See this Lightning Video.

Importing is also straightforward. You can drag and drop your ***.synexpz** file into the Processing tree of Synapse. See this Lightning Video.

Corpus Limitations

Because you are not using real hardware, there will be limitations on some closed-loop experiments because data cannot be output, and the inputs are limited to faking the signal in corpus.

Certain peripheral devices, such as IZ2 stimulators, RS4 data streamers, RV2 video trackers, cannot be emulated by Corpus. They can appear in the rig, but they will not do anything during runtime. See the Corpus User Guide for a list of supported devices.

How to Update Synapse

> **Important**
>
> **Licensing** - Synapse v96 has a new licensing model. See Synapse Licensing for more information and this FAQ guide.
>
> **New Installer Names** - 'Synapse Essentials' is now 'Synapse'. 'Synapse Suite' is now two separate packages: 'Synapse' and 'SynapseExtras'. The proper packages are automatically installed for you during the update process below.

Update to the latest version of Synapse by following these steps:

1. Download and install the latest "TDT Drivers/RPvdsEx" from the downloads page on our website. This installation will require a reboot.
2. Update the microcode on your devices. (Ensure the RZ system is connected and turned on)
   a. Open **zBUSmon** (black icon on the desktop or in C:\TDT\zDrv3).
   b. If you see the **Update All Devices** button, skip to the next step. Otherwise:
      i. Hold down the Shift key and right-click on the RZ in zBusMon.
      ii. Select **Program RZn_n**.
      iii. Click the **X** to close the **System3 Device Programmer** dialog. This resets the RZ as a **G21** device.
      iv. The **Update All Devices** button is now available.
   c. Click the **Update All Devices** button.
      - A time warning will be displayed. RZ processors may take up to 40 minutes (five minutes per DSP). If your system contains several devices this process could take significant time.
   d. Click **Yes** to continue.

   > **Important**
   >
   > The PC should not be used for other tasks while devices are being reprogrammed.

      i. If programming an **RX** device, a message will be displayed. Press and hold the **Mode** button on the front panel of the RX device and then click **Retry**. Release the

Mode button when the front panel of the RX device displays `Firmware: BLANK` or `Firmware: Burning`.

ii. When programming has completed you are returned to the zBUSmon window and the driver version should be displayed in the device diagram.

3. In Synapse, go to Menu → About → **Check for Updates** in Synapse. This will automatically download and install the latest version of Synapse.
4. Review the Release Notes for this version.
5. You are ready to begin using Synapse!

Contact TDT after installing to get your license key. Follow the instructions on the Licensing dialog. If you have any questions or problems during your update, contact support@tdt.com.

Release Notes

v100

Major Changes and Improvements

New Hardware!

- **iAn BioAmp Recording Module** - The iA4 Differential Bioamp provides up to four channels of true differential acquisition, ideal for long duration recordings without worrying about battery life. The iA16 / iA32 Bioamp records 16 or 32 channels of single-ended or true differential acquisition.
- **iD2 Digital Headstage Interface Module** - The iD2 records up to 256 channels from Intan-based headstages and streams the raw data back to the computer and straight to disk via USB 3. You can also visualize all channels during recording and select up to 16 for real-time processing.
- **Real-time Camera Tracking Added to iV2** - The iV2 Video Interface interface module now supports real-time subject tracking. This also requires a hardware update to the iV2. Contact TDT Support to coordinate the replacement and for help getting started. The iV2 replaces the RV2 Video Tracker.
- **iV1 Video Capture Module** - The iV1 is a low-cost solution for perfectly synchronized frame acquisition for subject monitoring. The iV1 replaces the CamHal Software USB Camera.
- **RS3 Data Streamer** - The RS3 is a DSP for your RZ2 processor that streams raw data directly to the PC via USB3. The RS3 replaces the RS4 Data Streamer.

Miscellaneous Improvements

iConZ Sampling Rate Unlocked

- The iConZ can now operate at up to 50 kHz sampling rate.

Better stability in USB3 Interface initial connection

- Affects iConZ, UZ3, and RZs with built-in USB3 interface that had issues first connecting to TDT software.

iV1/iV2 Synchronized to start of recording

- There is a new Preference called "Record Start/Stop Synchronization" that is by default turned on. This settings makes sure every device that needs time to start up is ready before the recording begins. For the iV1/iV2, this means the frames start precisely when the recording begins at time t=0.

Bug Fixes

- iMn Modules - Fixed PWM calculation
- Rig Editor - Fixed issue where very large rigs were unchecking some assets erroneously on exiting Rig Editor
- iMn Modules - Fixed Pynapse auto-complete options when using a mixture of Triggered and Strobe-controlled outputs
- Pynapse
    - Allow user to rename gizmo in Processing Tree
    - Allow larger range of values for integer Outputs
    - Fix Controls name length issue
- SynCon - Fix bug when launching with no rig
- iCon - Shorten epoc automatically generated names to avoid collision

Important

Legacy USB Drivers - Legacy USB drivers now require that Core Isolation is disabled on Windows 11. See Tech Note 2301 for more information.

Version 100-r54279 (2025/04/15)

New Hardware!

- **iX7 Integrated Photometry Interface** - iCon module for fiber photometry with integrated optical manifold
- **iConQZ Behavioral Controller** - Quad processor behavioral interface
- **RZ11 iCon Processor** - RZ processor with built-in iCon2

Miscellaneous Improvements

- Synapse
  - Allow File Stim and Ultrasonic File Stim gizmos to have negative amplitudes for inverted waveforms
  - Added iDn channel map files for ZD headstages
  - Updated iMn UI for clarity
  - Allow iDn to stream higher sampling rate than processor
- SynCon
  - Remove 6k system rate restriction
  - Add Single Unit package
- zBusMon
  - Added internal firmware version number to DSPs

Bug Fixes

- Synapse
  - Fix two CamHal crash
  - Fix iDn impedance check
  - Fix iDn focus channel table size
  - Fix IZV single channel impedance check
  - Fix iVn video file creation on first recording
  - Fix rare error creating iVn video files
  - Show User-defined parameter widgets in Pulse Train gizmo
  - Fix output subtraction in Fiber Photometry Gizmo
  - Fix load indicator when using USB3 interface

v98

Important

Windows 11 - Synapse v98 now supports Windows 11. See Tech Note 2301 for installing Optibit interface drivers.

Major Changes and Improvements

New Hardware and Gizmos!

- **iCon Behavioral Controller** - Supports new iCon-Z behavioral interface and new iCon modules (all iMn modules and iX6)
- **Ultrasonic File Stimulation Gizmo** - New gizmo for presenting custom stimuli from files at up to 200 kHz sampling rate
- **Python Coding Gizmo** - Major updates to the Controls interface, allowing more flexibility in designing the runtime UI

Changes to Gizmo Defaults

- Fiber Photometry clip threshold is now 9.0 V
- Smooth Scrolling is disabled by default
- `Period` and `Width` parameters for iM PWM outputs have been replaced with `Base Frequency` and `Duty Cycle`. The Pynapse `setPeriod` and `setWidth` API functions have been replaced by `setDutyCycle`

Miscellaneous Improvements

Electrical Stim Driver Gizmo

- Added PulseAct output signal

Bug Fixes

- Ultrasonic Stimulation - fixed the max duration (1e7 ms)
- Channel Mapper - ZCA-OMN32 map was inverted
- Pulse Train Generator, Audio Stimulation, Electrical Stimulation, Electrical Stim Driver, File Stimulation - Fixed Period parameter epoc storage when using jitter
- PCA Sorter - Fixed monitor compatibility with RZ6 DACs
- IZV Hal - Fix for using four sub-stimulators and multiple switching headstages
- User Gizmos - Fix when using long user gizmo names
- File Stimulation - The file "ID" parameter maximum value updates when the file list changes
- RZ Digital I/O - Allow multiple gizmos to read digital inputs without warning

Version 98-r51248 (2023/04/19)

Miscellaneous Improvements

- Synapse
  - Improved support for iCon iX6 device
  - Improved the Web Updater
- Pynapse
  - Minor Pynapse Controls layout updates
- ActiveX
  - Added support for UZ3 interface

Bug Fixes

- Synapse
  - Fixed PWM resolution selection on iCon iMn devices (removed PWM from iM10)
- Pynapse
  - Fixed input buffer reads
- Fiber Photometry Gizmo
  - Writes to Frequency and Level from SynapseAPI properly update the UI widgets

Version 98-r51739 (2023/07/23)

Miscellaneous Improvements

- Synapse
  - Added support for iCon iV2 video capture device
  - Added support for PS2 photosensor
  - Improved support for iCon iX6 device
- OpenScope
  - Added video playback for iV2

Version 98-r51878 (2023/08/21)

Miscellaneous Improvements

- Synapse
  - Fixed iV2 bug when creating reference videos for two cameras
  - Fixed iV2 Standby Mode support
  - Hide Dac-2 options for iMn devices with one DAC output
- zBusMon
  - Fixed crash issue when updating multiple RZ devices

Version 98-r51891 (2023/08/23)

Miscellaneous Improvements

- Synapse
  - Fixed Electrical Stim Driver switching headstage padding when Count = 1

Version 98-r52849 (2024/08/15)

Miscellaneous Improvements

- Synapse
  - Added support for iA4 Differential Bioamp Module
  - Allow Ultrasonic Stim to have negative amplitudes for inverted waveforms
  - Fixed Electrical Stim Driver epoc storage when "Local Ground" is selected
  - Fixed issue with Persistence UI control
  - Fixed OpenGL plot offset and window sizing issues
  - Fix SynapseAPI getSystemStatus return value
  - Add support for newer Python libraries in Pynapse
  - Fixed PZD issue when AC Coupled was unchecked

v96

Important

Licensing - Synapse v96 has a new licensing model. See Synapse Licensing for more information.

New Installer Names - 'Synapse Essentials' is now 'Synapse'. 'Synapse Suite' is now two separate packages: 'Synapse' and 'SynapseExtras'.

Major Changes and Improvements

New Hardware and Gizmos!

- **iCon Behavioral Controller** - Support for the iCon to interface with behavioral hardware components
- **Python Coding Gizmo** - Major updates to Pynapse gizmo for direct iCon integration, with new plotting, experiment design, and progress tracking features added. If you have Pynapse experiments designed in v95 please contact TDT.
- **Switching Headstage** - Support for the new ZC-SW switching headstage.
- Added online impedance checking for PZ5 and Subject Interface (IZV, PZA, and PZD).

- New User Gizmos for simple but common processing tasks in the Custom → TDT gizmo category.

Changes to Gizmo Defaults

- The 'Tick' store is enabled by default in the RZ options and plotted
- Audio Stimulation and Ultrasonic Stimulation Gizmos - the Phase Sync option is 'Sync to Stim' by default
- Audio Stimulation - unit text for WaveAmp parameter is now 'dBV' to accurately reflect the units
- Fix duplicate epoc names across multiple RZ/RX digital IO outputs
- Fiber Photometry Gizmo - Driver-3 defaults to 530 Hz for better 50 Hz power line compatibility.
- Medusa4Z is default RAn HAL on the Rig Editor auto-detection.

Miscellaneous Improvements

Rig Editor

- Rig editor properly auto-detects PZ5 configurations

Experiment Dialog

- Can now sort experiments by Date Created and Date Modified

Electrical Stim Driver Gizmo

- EStim Driver can now save parameters 'On Pulse'

New RPvdsEx Components

- New MCiShift, MCiAnd, MCiOr, AndOr components

Pulse Train Generator Gizmo

- Added checkbox to mute output at runtime

OpenScope

- Use the new View → 'Show Epocs on Time Navigator' menu option to hide epocs from timeline for improved graphical performance while replaying long recordings.
- AVI file names no longer need the entire block name as a prefix. Load video files named Cam#.avi and Vid#.avi files from the network.

SynapseAPI

- SynapseAPI now returns error codes if a call fails
- Writes to User Gizmo parameters are sent right away

MRI Processing Gizmo

- MRI pro and Neural Stream processor check device sampling rate to match to store rate

PZ Hardware

- PZA now has clip + activity LED options, like PZ5 Analog
- PZD now has headstage detect LED options, like PZ5 Digital

Fiber Photometry Gizmo

- The Fiber Photometry Gizmo has an option to raise the lowpass filter cutoff for TEMPO (voltage sensor photometry).

Parameter Sequencer

- Click any row or header during runtime to set that sequence or manually present that row.

Tank Files

- The StoresListing.txt file saved in the block contains more summary information on data stores in the recording.
- Channel map CSV file(s) are included in TIN file in the block folder.

General

- Added default Pynapse environment as a setting in the Preferences dialog
- Error messages during mode change are more user-friendly
- Synapse in-app help link goes to documentation website first

Bug Fixes

- Channel Mapper - doesn't forget upper map channels when lowering the channel count on the input
- Pulse Train Generator - was not applying correct initial period at run time
- If the flow plot settings don't save properly, they are automatically reset on next recording
- Allow multiple instances of MRI Processing Gizmo in same experiment
- RZ Port C grouping bug fix when not inverting inputs
- IZV - Fixed 'Serial' mode across multiple banks
- Parameter Sequencer - Fixed 'New' button for Parameter and Sequence tabs
- Pulse Generator - fixed min/max control limits

Version 96-r48847 (2022/02/04)

Miscellaneous Improvements

- Synapse
    - Added RS4 runtime status monitor
    - Temporary files are permanently deleted, not sent to Recycle Bin
    - Experiment list view is now the default
    - Auto-detect PZn on DSPM
    - Auto-detect SIM, iCon, or PZn device on DSPQ optical card
- Pynapse
    - Automatically install a default python package
- Fiber Photometry Gizmo
    - Added storage sampling rate slider for demodulated and calculated signals
- Electrical Stim Driver Gizmo
    - Added 'ChargeParams' parameter to adjust all biphasic waveform parameters (T1, Td, T2, L2) in a single call
- User Input Gizmo
    - Updated SynapseAPI trigger behavior, 1 to turn on 0 to turn off

Bug Fixes

- Synapse
    - Fixed MCFIR component calculation
    - Fixed issue with Clear Session
    - Fixed PZn analog subamp impedance checking for sub amp 2 or higher
    - Fixed Parallel mode in IZV
- Fiber Photometry Gizmo
    - Fixes visual issue with offset spin box step size
    - Fixed issue with setting difference source in calculated outputs
- Pulse Train Generator Gizmo
    - Fixed issue with 'Timer' mode
- Electrical Stim Driver Gizmo
    - Fixed SynapseAPI trigger in strobe limited mode
- Fixed v96 RXn microcode
- Corpus
    - Fixed tone generator component
- OpenBridge
    - Fixed issue with NEX export
- Pynapse
    - Fixed issue with input buffers

Version 96-r49154 (2022/02/24)

Bug Fixes

- Synapse
    - Fixed iM10 White Noise list indexing

v95

Major Changes and Improvements

New Gizmos!

- **Python Coding Gizmo** - A gizmo for tightly integrating Python coding into your Synapse experiment. With Pynapse you can:
    - Control your experiment flow
    - Build and deliver stimuli
    - Run complex behavioral paradigms
    - Do custom analysis and visualization
- **Pulse Train Generator** - Create simple or complex user-defined pulse train waveforms. Trains can be stacked or parallel. Use this gizmo for directly and dynamically controlling optogenetic stimulation.

Miscellaneous Improvements

Fiber Photometry Gizmo

- Optimized Fiber Photometry Gizmo circuit to reduce processor usage.
- Increased max bleaching current to 1000 mA.

Bug Fixes

RZ10 and RZ10x Fiber Photometry System

- Some LUX LEDs have a larger leakage than expected when using the RZ10x at the 500 mA or 1000 mA range for stimulation. This doesn't affect fiber photometry users who run in the 50 mA and 200 mA ranges. The control signals are updated in v95 to resolve this.

PDF Help Link

- Internal help link in Synapse now compatible with Adobe Reader 2020.

File Stimulation Gizmo

- Fixed issue where first file in list was stopping one sample short during presentation.

Version 95-r43524 (2020/07/10)

Bug Fixes

- Fiber Photometry Gizmo - Fixed max current setting during automated photobleaching.

Version 95-r43718 (2020/08/04)

Miscellaneous Improvements

New User Gizmos

Version 95-r44132 (2020/09/24)

Bug Fixes

- Electrical Stim Driver Gizmo - Fixed StimSync when Count=1. It was previously 1 sample too short. When using a Subject Interface, this would switch into inter-stim action mode too early.

v94

Major Changes and Improvements

New Hardware!

- **Support for the RZ10x Fiber Photometry System** – The RZ10x has integrated Lux LEDs, photosensors, and power meters for a more streamlined fiber photometry setup
  - RZ10x System 3 Manual
  - Synapse Manual

Fiber Photometry Gizmo for RZ10x

- New gizmo features for RZ10x specific fiber photometry experiments. Now includes auto-detect of integrated LUX hardware, power meter readings, and photobleaching/ experiment timing controls (What's New document for getting started)

Version 94-r42320 (2020/02/18)

Miscellaneous Improvements

SynapseAPI

- User Gizmos and PA5 parameters are written much faster

Fiber Photometry gizmo (for non-RZ10x users)

- Removed 'Auto-Enable' checkbox for each driver. There is now one master control to enable lights in Outputs and Data Saving.
- Raw stores (Fi1r) are now split up into photosensor (Fi1r) and driver (Fi1d) stores

Synapse Preferences

- Added setting to select new hardware state behavior when making new experiment: fresh, keep, prompt.

Electrical Stim Driver gizmo

- Waveform store disabled by default. Default target changed to IZV10.

Epoc Event Storage gizmo

- Default setting is now 'Strobe Input'

File Stimulation gizmo

- Added TXT and CSV file options

Bug Fixes

- Fixed Subject Interface closed-loop compilation errors and erroneous timing errors on mixed SIM boards
- Fixed Neural Stream Processor issue with adjusting filter frequencies at runtime
- Electrical Stim Driver gizmo
  - Fixed monopolar stim – stim duration was fixed at 2 ms
  - Fixed parameter table – adjusted to V instead of mV when IZ2 target in Voltage mode
  - Fixed channel count for >24 channels to IZ2.
- Fixed Corpus Counter and DSPP + PZ5 emulation
- Synapse Feedback button URL link is no longer broken
- Fixed Pulse Generator gizmo API control of 'Enable' parameter
- Adjusted Ultrasonic Stimulation gizmo max duration based on processor sampling rate

v92

Major Changes and Improvements

New Gizmos!

- **MRI Recording Processor** - MRI artifact rejection filters for cleaning continuous single unit and LFP signals during magnet switching or other large sources of artifacts.

New Hardware!

- Support for the **Subject Interface Module** (SIM) – SIM integrates new high-resolution stimulation with PZ5 analog and digital recording cards
  - System 3 Manual
- Support for the **Medusa4Z** - A new high-resolution four-channel preamp for low impedance recordings.
  - System 3 PDF link

Gizmo Slides

- Gizmo Slides are a user dialog in Synapse that provides an overview of what each gizmo does and the common connections they have

Legacy Circuits now support Gizmo Control Widgets and Storage Gizmos

- Add gizmoControl widgets and Storage Macros in Legacy circuits to control circuit parameter tags and see stored data in Synapse

Parameter Sequencer Custom Timer

- Added timer column to Parameter Sequencer to have explicit control and complete flexibility of when each presentation occurs

Version 92-r40727 (2019/06/13)

Miscellaneous Improvements

Electrical Stim Driver

- Added API parameters for biphasic wave construction - T1, Td, T2, L2
- Electrical Stim Driver detects SIM voltage mode and changes units accordingly

Parameter Sequencer

- Faster pasting and more options in CSV table
- Fixes to calculator and table
- Can change sequencer headers from 'Seq-N' to custom identifier

Lab Rat

- Fake Brain 'Stim Sync' now working with external and gizmo strobe input

OpenBrowser

- Added clip min/max parameters for EDF exporting to improve resolution

OpenExplorer

- 'Auto Plot New Block' option in Control menu automatically selects new block when in Track mode

Fiber Photometry Gizmo

- New defaults:
    - Low pass changed from 3 Hz to 6 Hz
    - Clip Threshold changed from 3.5 V to 7 V
    - Light driver DC Offset changed from 80 mA to 20 mA
    - Allow user to set required sampling rate if higher sampling frequencies are needed

Runtime Recording Notes

- Epoc timestamps for note events now show up on Flow Plot

Revision Log

- User can mark experiment versions as 'stable' in revision log dialog. Useful if you are continually editing an experiment but want to revert to a known good state.

File Stim gizmo

- 'FileList' parameter added. This returns list of loaded file names when called through API. This list also gets saved in the *.tin file of each recording block

Compiler Optimizer

- Now On by default in new installations
- Increased 'Processing' slider limit when enabled

Keyboard shortcut

- can use F7 hot key to compile in Synapse (shift + F7 to revert)

Stim Gizmos

- Increased maximum values for several options in the parameter tables

Bug Fixes

- Fixed Electrical Stim Driver gizmo pulse timing - there were two extra samples at the end for Td and T2
- IZ2H/IZ2MH voltage scale factor fixed
- Corpus clock inaccuracy fixed
- The extra low pass filter option 'Add DSP filter' to remove high frequency digital noise on digital banks might not have worked with mixed analog/digital PZ5 configurations or multiple logical amplifiers
- RPvdsEx components ArtReject, Counter, TSlope, and IZ2 emulator fixed in Corpus
- AudioStim/UltrasonicStim set phase to ±180 degrees fixed
- Persistence changes in Standby mode is now saved to the database
- Fixes for PO8e HAL integer streaming modes
- Epoc Storage gizmo – fixed value change mode for integer inputs

v90

Special Notes When Updating

- The LFP gizmo is replaced by the Neural Stream Processor gizmo and
- The Electrical Stimulation gizmo is replaced by the Electrical Stim Driver gizmo.

To see the deprecated gizmos in the list of available gizmos, select the option in Menu → Preferences → Show Deprecated Gizmos.

Major Changes and Improvements

New Gizmos!

- **Signal Accumulator** - Computes running sums and averages over a defined window, and allows advanced plotting and thresholding of these values
- **Electrical Stim Driver** - A more powerful version of the Electrical Stimulation gizmo that allows up to 4 independent stim patterns (voices) with built-in channel selector for controlling IZ2 stimulator.
- **Ultrasonic Stimulation** - A stripped down version of the Audio Stimulation gizmo designed to run at higher frequencies on the RZ6 processor
- **User Input** - Bring external digital inputs (e.g. button press) or software button presses into Synapse for marking timestamp events and controlling logic flow

Multiple run-time plots and advanced layout saving

- Useful for multi-subject work or splitting the plots across multiple tabs or monitors

New run-time plots added to Strobed Data Storage gizmo

- Add real-time heat maps, bar graphs, and snippet plots to your Strobed Data Storage

Added API access directly to gizmo ports, including RZ inputs and outputs

- Most gizmo outputs can be read directly using the API instead of going through a User Gizmo with SynapseAPI Access

Version 90-r39052 (2018/09/20)

Miscellaneous Improvements

Parameter Sequencer

- New Parameter and Sequence Generator lets you quickly build out tables of combinations and permutations.

Unary Processor

- Added a visual schematic to better understand the signal flow and how the parameters are used.
- When converting to logic signal, the floating point signals before the logic test are also available on the output. Example: If making an RMS threshold detector, you can visualize both the threshold crossings and the RMS value used for the detection.

PCA Spike Sorting

- Now runs at up to 100 kHz sampling rate. RZ6 users can combine high frequency audio stim and low channel count spike recording on one device.

MATLAB SDK link to SDK page

- Added functions to TDTbin2mat to apply digital filters (TDTdigitalfilter) and threshold continuous streams into snippets (TDTthresh).
- Added 'COMBINE' option to TDTbin2mat which can merge snippets of data that was stored in Strobe Controlled mode using the Strobed Data Storage gizmo back into its constituent streams.
- Expanded STORE option to include cell array of strings.

Preferences

- New Compiler Optimizer option in Advanced tab removes unused circuit components during compilation.

Parameters

- Added right-click pop up dialog on parameter table values that allows easier parsing of input values, such as E-notation.

Epoch Event Storage

- Updated the UI and added 'On Change' mode that stores the value whenever a change is detected on the input value.

Selector Gizmo

- Added 'Logical Output' checkbox when selecting bit fields or sort codes. In Bit Fields mode, logic output is 1 if selected field's value equals selected sort code. In Sort Codes mode, logic output is 1 if selected field's value is nonzero.

Neural Stream Processor Gizmo

- Output link is now always enabled.

Pulse Generator Gizmo

- Added 'Minimum' Duty Cycle option which makes output pulse a single sample.

Oscilloscope Gizmo

- Run-time settings are kept when switching to Idle, including threshold(s) and plot range, so your design and run-time interfaces are always synchronized. The new Accumulator gizmo also does this.

RS4

- Optionally allocate more DSP memory when streaming very high channel count data.

Bug Fixes

- Neural Stream Processor high pass filters below 2 Hz were not applied correctly (no high pass filter was applied).
- Fixed 'Save to Disk' option in all gizmos that do data storage so that nothing about the store is saved to disk when this option is disabled.
- Fixed cases where Corpus becomes unresponsiveness
- Tetrode Spike Sorting - Fix when deleting sort codes
- Better handling of very high and very low screen resolutions
- Bug fixes for some lesser used Corpus emulated DSP components
- Fixed Mapper crash bug when using multiple custom maps
- Fixed Unary Processor multi-channel convert to logic

v88

Special Notes When Updating

- Make sure all gizmo names are less than 15 characters long.
- Individual Referencing Mode in PZ5 maps only the odd channels to the Amplifier output.
- LFP gizmo will be replaced by Neural Stream Processor gizmo in the next release.
- Stimulation gizmos no longer fire automatically when attached to #Reset signal.

Major Changes and Improvements

New Gizmos!

- **Neural Signal Referencer** - Removes common signals from a multi-channel stream of neural signals. Can do single or multi-channel referencing on all channels or independent sub-groups of channels.
- **Timer** - Measures the elapsed time between logical events or calculates the frequency of events.
- **Neural Stream Processor** - Filters and stores single or multi-channel biopotential waveforms. Replaces the LFP Gizmo.
- **Pulse Generator** - Creates a simple user-defined pulse train. Useful for turning on/off the fiber photometry light driver outputs or controlling optogenetic stimulation.
- **Delay** - Adds a fixed or dynamic delay to any signal.

Support for UZ3 Fiber Optic to USB 3.0 interface.

Synapse Lite and support for Lab Rat platform (LR10).

Version 88-r38174 (2018/04/23)

Miscellaneous Improvements

- New interface button on upper right to provide direct feedback to TDT.
- New notes interface in RZ and RX devices. Save string notes and timestamps into the tank and a readable Notes.txt log in the block file. Can also create notes through the SynapseAPI.
- StoresListing.txt file in block folder contains information about what gizmo created each store.
- Epoch store gizmo can now be triggered on value change.
- Electrical Stimulation gizmo shows a plot of the pulse and burst during design time.
- Parameter Sequencer lets you automatically start the sequence when the block starts and automatically store the recording when the sequence is done.
- Button presses and all SynapseAPI calls to write to buttons are passed directly to circuit with minimum delay.
- Each gizmo can be set to 'Fresh' persistence on a per-gizmo basis. When selected, the run-time persistence will be ignored and the default experiment values for that gizmo will be used every recording.
- User run-time layout can be loaded from pre-existing experiment run.
- User gizmo - gizmoControl macro allows duplication of controls for parameter tags inside iterate box.
- The RV2 has fewer communication delays.
- PCSort/BoxSort/TetSort can show just one channel/tetrode.
- 'Tick' store can now be disabled to save cycle usage.
- Experiments are lockable to prevent accidental edits.
- "Clean Storage" menu lets you remove old Preview history and empty tanks to clean up the database and hard drive.
- Stimulation gizmos have the continuous store enabled by default.
- Run-time channel map editing is disabled.

Bug Fixes

- Fixed phase sync in audio stimulation gizmo.
- LFP HP filter API setting was off by one.
- Can connect User Gizmos to SortCodes output links directly now.
- Fixed bug in Smooth Scrolling data plot when paused.
- Fix RVMap for invalid parameters.
- Allow Parameter Sequencer gizmo to have more than 10000 items.
- Make sure duplicate Fiber Photometry gizmos create unique stores.
- Fix off by one sample bug in File Stimulation gizmo.
- Fix for initial value when gizmoControl is a Logic type.
- Properly limit the ZD headstages to 25 kHz sampling rate.
- Fix for stimulation gizmos missing first trigger when in Strobe controlled modes.

v86

Major Changes and Improvements

Cluster Mode

- Using PO5c interface card, divide complex tasks across multiple workstations, screens, and users. Easily scale to 2,000 recording channels and beyond, all time-locked to a single master clock. Or split high channel count recording and custom stim paradigms across two machines.

Version 86-r36093 (2017/07/19)

Miscellaneous Improvements

- Data Plot
  - New smooth scrolling option
  - Better memory constraint handling
- PZn HAL
  - Added clip and activity LED checkboxes.
- LFP gizmo
  - Added option to turn off high-pass filter.
- PCSort, BoxSort, TetSort gizmos
  - Plotting routines are faster.
  - Snippet storage is more efficient in v86 microcode.
- SynapseAPI
  - Added getMemos function to read log files.
  - Added getGizmoParent function.
  - Updated getKnownBlocks function to return all known blocks in Synapse database.
- Updated Preferences Dialog to give user more control under the hood.
- Improved accuracy of automatic impedance checker for IZn HAL.
- User can export the Subject, Experiment, and User logs directly to CSV files. They are also accessible through SynapseAPI.
- Added PA5 HAL.
- Added menu options to delete empty tanks from disk.

Bug Fixes

- Fixed bug where low-pass filter was applied to PZ2 and PZ4 devices.
- Fixed Unary Processor preset order.
- Increased Parameter Sequencer gizmo interval precision and maximum value.
- Fixed bug in Sort Code output from BoxSort/TetSort gizmos when connecting to Sort Binner gizmo.
- Fixed RX8 analog I/O montage.
- Fixed RX/RZ grouped digital I/O inversion.
- Fixed PCSort unsorted vs outlier sort code identifier.
- Miscellaneous UI fixes.
- Warn user when changing name of gizmo so that proper persistence can load.

Upgrading to Windows 11

Windows 10 has an end of life date of October 14, 2025 (see https://www.microsoft.com/en-us/windows/end-of-support). As a result, many universities are requiring research machines on their networks to upgrade their existing OS to Windows 11 for security purposes. This document is intended to help you smoothly transfer your existing TDT software to your new OS or new machine.

> **Note**
>
> TDT recommends backing up all existing data and TDT folders to a backup hard drive or computer image before upgrading the OS of your computer.

While all TDT projects, which includes RpvdsEx circuits + ActiveX code, BioSigRZ, OpenProject experiments, and Synapse experiments, are forward compatible with all newer versions of TDT software releases, you will have to upgrade to v98 or later TDT software to stay compatible with Windows 11.

If you are upgrading to a brand new machine with no existing TDT software, you can easily download the latest versions of everything from our Downloads Page and proceed from there. Just be sure to also update the microcode on your hardware if updating your software to the latest version. If you need assistance with passwords or any part of the installation process, please contact support@tdt.com.

For Synapse Users

Synapse information is easily transferred because everything in Synapse (besides actual data and extra-database files such as user circuits) is saved into a database file. This contains all of your experiments, subjects, and runtime settings.

There is a folder `C:\TDT\Synapse\Backups` that contains, among other things, the last hardware rig you used (`.synrig`), a `UserSettings.ini` file, and the database file (`.db`).

To restore Synapse on a fresh computer:

1. Install TDT Drivers and Synapse on the new machine.

2. Replace `synapse-1.db` in `C:\TDT\Synapse` with the `.db` file from your `Backups` folder.
3. Import the rig file (`.synrig`) via Menu -> Edit Rig -> Import.
4. Move your older `UserSettings.ini` file into

   `C:\Users\{username}\AppData\Local\Tucker-Davis Technologies\Synapse`

   to restore all of your Menu -> Preferences settings.
5. Additional important folders to migrate include: `C:\TDT\Synapse\UserCircuits` for any experiments with user gizmos or legacy circuits, `C:\TDT\Synapse\MapFiles` for any experiments with map files, `C:\TDT\Synapse\ParFiles` for experiments with Parameter Sequencer lists, `C:\TDT\Synapse\StimFiles` for any experiments that use fStim and stim files, and `C:\TDT\Synapse\CamRefs` for any experiments that used an iVn Video Capture iCon module.

   Please copy these folders, as is, into the `C:\TDT\Synapse\` directory (replacing the existing fresh folders) so synapse can recognize any appropriate files if there are dependencies.

For OpenEx Users

Projects folders can be zipped and easily transferred to different computers. The only thing you will have to do is make a new Tank and Block location for the data to be stored if the folder paths are different on the new machine.

For BioSigRZ Users

Biosig configuration files (`.acf` files), calibration files (`.tcf`), and SigGen stimulus files (`.sig` files) are easily ported into a new installation of BioSig. Place `.acf` files into `C:\TDT\BioSigRZ\Configs`, `.tcf` files into `C:\TDT\BioSigRZ\TCF`, and `.sig` files into `C:\TDT\BioSigRZ\SIG`.

For ActiveX Users

ActiveX code and associated RPvdsEx circuits are readily used on newer versions of TDT software.