Synapse Manual
Copyright
©2016 Tucker–Davis Technologies, Inc. (TDT). All rights reserved.

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of TDT.

Tucker–Davis Technologies
11930 Research Circle
Alachua, FL 32615 USA
Phone: (+1)386.462.9622
Fax: (+1)386.462.5365

Notices
The information contained in this document is provided “as is,” and is subject to being changed, without notice. TDT shall not be liable for errors or damages in connection with the furnishing, use, or performance of this document or of any information contained herein.

The latest versions of TDT documents are always online at www.tdt.com/support.htm.
Synapse Manual Table of Contents

Synapse: Next Generation Performance Today ........................................................................................................ 1

Part 1: Getting Started with Synapse
Before You Begin ............................................................................................................................................................ 9
Launching Your First Experiment ...................................................................................................................................... 13
Using Gizmos to Build an Experiment .......................................................................................................................... 23
Managing Data for Your Lab ........................................................................................................................................ 29
Managing Users and Subjects ...................................................................................................................................... 33

Part 2: Synapse Fundamentals Reference
Hardware Configuration .......................................................................................................................................................... 43
Designtime Reference ...................................................................................................................................................... 51
Runtime ........................................................................................................................................................................... 63

Part 3: HAL Reference
Hardware Reference ............................................................................................................................................................ 71

Part 4: Gizmo Reference
Logic
State Maker ........................................................................................................................................................................ 93

Neural
Local Field Potentials (LFP) .............................................................................................................................................. 101
PCA Spike Sorting .......................................................................................................................................................... 105
Tetrode Spike Sorting ...................................................................................................................................................... 115
Box Spike Sorting .......................................................................................................................................................... 129
Sort Binner .................................................................................................................................................................... 137
Fiber Photometry ............................................................................................................................................................. 139
Routing
Mapper ......................................................................................................................... 147
Injector ......................................................................................................................... 151
Parameter Manifold ....................................................................................................... 153
Selector ......................................................................................................................... 157
Merger ......................................................................................................................... 159

Signal Conditioning
Artifact Blocker ............................................................................................................ 163
General Purpose Filter .................................................................................................. 167
Unary Signal Processor ................................................................................................. 169

Stimulation
Using Bounded Parameters ......................................................................................... 175
Parameter Sequencer ..................................................................................................... 181
Audio Stimulation ......................................................................................................... 187
Electrical Stimulation .................................................................................................... 195
File Stimulation ............................................................................................................ 201

Storage
Epoch Event Storage ..................................................................................................... 211
Stream Data Storage ..................................................................................................... 213
Strobed Data Storage ................................................................................................... 215

Visualization
Oscilloscope .................................................................................................................. 221

Part 5: User Gizmos
Creating User Gizmos ................................................................................................... 229

Index .................................................................................................................................. 235
Synapse: Next Generation Performance Today

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 25 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 puts a layer of abstraction (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.

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.

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.
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.

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. OpenEx users will find many of these runtime interfaces familiar, as many got their start as SpikePac tool-sets. The interfaces have been improved and expanded for Synapse.

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.

Next Generation Performance Today

For current OpenEx users, the upgrade path is fast and painless. Many of the components of your current OpenEx experiment are replaced by updated TDT Gizmos. Any parts of the experiment that don’t match existing built-in tasks will be easily ported into Synapse through user designed gizmos.

At TDT we take pride in leading innovation, but we know innovation doesn’t mean continually scrapping what you have and starting over. When developing tools for your research, building on what we’ve learned and asking fresh new questions yields solid results. We take the best of today’s technology and springboard forward to
deliver tools you can build on. Synapse gives you the power to design, manage and collect data from your experiments using next generation technology, today.
Part One: Getting Started with Synapse
Before You Begin

Installation

Synapse can be installed from the TDT Installation CD or downloaded from the TDT website as part of the Synapse Essentials or Synapse Suite bundle.

TDT Drivers should be installed first.

**Synapse Essentials** includes:

- Synapse
- TDT Drivers
- TTankMin
  - TTankX
  - OpenScope
  - OpenBrowser
- OpenBridge
- Synapse API

**Synapse Suite** includes:

- Synapse Essentials (above)
- OpenExplorer
- OpenController
- OpenSorter

PC Requirements

The recommended operating systems for all TDT systems is Windows® 7 or 10.

**PC Hardware Requirements:**

- 2.0 GHz or faster processor (Intel® Core™2 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)
• CD-ROM drive
• Full height PCIe slot

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:

- .synexp 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).

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.

Organization of the Manual

This manual will help you get started using Synapse software and serve as a long-term source of reference information.

Getting Started with Synapse

This section provides a high level walk through of building and running a basic experiment along with an introduction to many important Synapse concepts and techniques.
Synapse Fundamentals
A reference is provided for each phase of using Synapse. Reference guides include detailed references for windows, menus, dialog boxes, and settings.

HAL and Gizmo Reference Guides
References are provided for each Synapse hardware configuration (HAL) and gizmo. Reference guides include configuration options and runtime interfaces.

Custom User Gizmos
A detailed guide to building and adding custom gizmos is also provided for customers transitioning from OpenEx.

Using the Manual
This manual uses the following icons to alert you to important content:
- Tip
- Technical Details
- Core Concept
- Video Demonstration Available
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.

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.

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.
Detected devices in your hardware configuration are displayed in a hierarchical diagram. Configuration settings for the selected (highlighted) device are displayed in the area to the right.

**Enabling Devices**

There are some device and configuration information that Synapse can’t automatically detect. In the illustration above, based on the equipment detected, such as the RZ2 and DSP-I, an IZ2 stimulator is likely to be a part of the attached system. Synapse added the IZ2 to the diagram, but it hasn’t been enabled. You would need to click the check box to enable it.

You can also enable or disable devices to control whether they are automatically added to the Processing Tree in new experiments. Only enabled devices are added by default. Disabled devices can still be added later and enabled manually in the Processing Tree. See “Disabled Devices and the Processing Tree” on page 45 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.

- Medusa PreAmp Add RAn
- USB Camera Add CAM
- UDP Interface Add UDPRcv and Add UDPSend

**Configuring a Device**

Some devices require that you configure some aspect of the device, even if it is automatically enabled.
The most common item you will need to configure is your PZ amplifier. You’ll need to select the number of channels available in your amplifier. In the rig, your device settings should exactly match your hardware. You’ll be able to reduce the number of channels actually used or make other experiment-specific changes to settings in individual experiments.

When everything is configured, you can update the rig by clicking OK to commit the changes and return to the Processing Tree.

In the illustration above, the UDP device icon has been added but is displayed with a warning indicator. The UDP requires an input source. It can be set now or it can be set later when the desired input is available.

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 on the main menu.
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’s 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’s unable to do so, alert symbols are added to the problem elements in the Processing Tree.

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**

In the new experiment, two task blocks called ‘gizmos’ are added to the Processing Tree beneath the PZ amplifier: one for LFP filtering (Lfp1) 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” on page 89.

**Naming the Experiment**

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

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 design-time 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 design time 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” on page 33.

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

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle Mode</td>
<td>Devices are not loaded and are not running.</td>
</tr>
<tr>
<td>Preview Mode</td>
<td>Data is acquired, but deleted after the recording ends.</td>
</tr>
<tr>
<td>Record Mode</td>
<td>Data is acquired and stored to the data tank permanently.</td>
</tr>
</tbody>
</table>
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 (rather than after).

The basic window includes a plot for each type of data being stored and 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” on page 63.

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

The second tab in the template is derived from the PCSort 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” on page 105.

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 sessions. Any changes made to a setting is logged in the database as well as the user and/or subject information. All of these values are retained and saved as part of a history of the experiment. This database of experiments and history of it’s past states (collection of settings for a user and/or subject at a given time) support 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.

- **LOCK**
  - Locks selected persistence.

- **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 for the current experiment (regardless of subject or user).

- **FRESH**
  - Uses the settings in the designtime Options area and clears any past runtime settings.

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

**User History**

The History Window is available from the button, in the Persistence section of the command bar.
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. When you return to Idle, you can return to the default layout using the **RESET 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” on page 29.
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.

TDT Gizmos (partial list)

Synapse’s designtime experiment interface in 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.

Options
Configuration options for the item selected in the 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 appear 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 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” on page 43.

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 Processors 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.

Multiple tasks can be added to any branch, in parallel or in a chain, with tasks ordered according to data or signal flow.
In figure A, a stream store gizmo is added to a filter. So, filtered waveforms are stored. In figure B, 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.

Synapse adjusts the available options as you work so that only relevant choices are available. 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.

**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.

**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).
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](image)

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” on page 229 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

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” on page 89.

**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” on page 57 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

Default Tank Path

Tank and Block Naming

Synapse provides a structured but flexible automated solution for tank and block naming that is 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 experiments and blocks of data are automatically named based on subject for each recording session. For example, an experiment named “LFPbase” run for the first time on October 8, 2015 at 10:51:34 is named:

LFPbase-151008-105134
ExperimentName-yymmdd-hhmss
Time is the onset value.

Default Tank Name

The blocks are named based on the subject. Three blocks might be:

MouseC5-151008-105134
MouseC6-151008-110526
MouseC7-151008-112049
SubjectName-yymmdd-hhmss
Time is the onset value.

Default Block Names
Note: 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.

Accessing Stored Data

Synapse uses the DataTank format, originally designed for TDT’s OpenEx Software Suite. Data can be viewed using OpenEx applications, including OpenScope and OpenExplorer, or accessed for analysis via Matlab using OpenDeveloper methods. As discussed above, a 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 beneath the Persistence selections on the command bar.
To open the History window:

- Click the User Search/Browse button.

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 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.
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” on page 29, 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 their 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**.

2. 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.
   - **Runtime Window Layout Check Box** – select to make layout unique per user.

Privacy

The purpose of privacy 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. It’s important to understand that 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 state 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.

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.

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 Window Layouts, Runtime, and Persistence, 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 a Subject:

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

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.

The subject is added.

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.

Synapse makes it easy to add many subjects in advance of your experiments. Instead of choosing NEW on the Subjects menu, choose MORE. The Subjects Window will be displayed and you can add new subjects one after another without returning to the main Synapse window.

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.

Using Persistence with Users and Subjects

The Persistence runtime interface was introduced in “Launching Your First Experiment”—“Using the Runtime Interface” on page 20. 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 clearing all settings back to the default, whenever selected.

**Best and Last**

By default, Synapse uses the BEST option, which applies the most recent settings for the current experiment, current subject, and current user. While the LAST option uses the most recent settings for the current experiment, regardless of user or subject.

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

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.

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.
The difference between LAST and BEST gains another layer of significance when users are added. If two users are running the same experiment, but one is using a different set of parameters, it will be important to stick with the BEST persistence. Using BEST ensures that user Kim does not accidentally run the experiment with user Smith’s settings.

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.

Windows Layout

It is important to note that the arrangement of windows, called the ‘Layout’, is tied to the user separately from the persistence settings. Layout and persistence are independent, though because they both rely on users it might seem otherwise. Either the parameter settings or window layout can be cleared without clearing the other.

To clear the Window Layout:

- Click Reset Layout at the bottom of the Persistence area on the command bar.

To clear the parameter settings:

- Click Fresh in the Persistence area on the command bar.
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” on page 13 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. This chapter takes a closer look at the auto-detection process and what to do if your hardware changes or you need to run the experiment with another system.

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 Tree Diagram
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” on page 48.

The Detect feature currently 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. You need to review the tree to verify that it correctly represents your system and to enable/disable, add, or configure devices as needed.

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. You will need to enable it before you can use it.

Enable and Disable Devices

Each item in the tree includes an enable check box, a representational icon and a label.

To enable/disable a device:

- Select or clear the Enable check box to the left of the device icon in the tree diagram.

Because the diagram is hierarchical based on connections, disabling a device necessarily disables devices below it on the branch.

Enable Check Box Key

- □ Device (and any nested devices) disabled
- ✔ Device (and any nested devices) enabled
- ✔ Device enabled, some devices in the branch are disabled
To enable or disable all items on the branch:

1. Click the check box of the device at the top of a branch.

2. If needed, repeat until a check mark (enabled) or no check mark (disabled) is displayed in the check box.

You can also enable or disable devices to control whether they are automatically added to the Processing Tree in new experiments. Only enabled devices are added by default.

Disabled Devices and the Processing Tree

Disabled devices can later be added to an experiment in the Processing Tree using the ADD HAL command on the EXPERIMENT BUTTON menu. 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

In the hardware tree, all DSPs are represented with a group icon and connected beneath the RZ processor where they are housed. Individual icons are then nested below. This generally places them at or below the level (in the tree) as external devices, such as amplifiers. This represents that some specialized DSPs boards include an optical interface that serves (in addition to the standard RZ inputs and outputs) as a communication connection to a peripheral device.

DSP labels provide additional information about the type of DSP and its associations. The label takes the form $\text{DSPX#}$ where:

- $X =$ device type to which the DSP connects, such as amplifier or 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

$\text{DSP Type/Device Connection Key}$

- S RS4 data streamer
- V RV2 video processor
- I IZ2 electrical stimulator
- U PO8e streamer
- P PZ amplifier
- Q Quad Core processor

$\text{Note:}$ 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) or number of channels.

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.

Amplifiers are a special case, in that they can be connected to the RZ’s standard optical connection and/or they can be connected to a DSP-P card. In the illustration below, one PZ5 amplifier is connected to the RZ2’s standard amplifier optical input port and a second PZ5 amplifier is connected to a specialized DSP-P card.

---

**Add/Remove Devices**

If a device has been added or removed from your system you can add or remove the device to/from the tree diagram. If you are adding a device you must first determine where the device should be connected in the communication diagram. For example: a PZ5 is typically added to an RZ processor.

**To add a non-processor device:**

1. Right-click the device with which the new device will communicate.
2. Click the desired device type on the shortcut menu.

Processor devices can also be added using the Detect button, however, any configuration of non-processor devices can be lost. To prevent loss of existing configuration the configurations must be merged.
To add a processor using Detect:

1. Ensure the Merge Previously Saved Configuration check box is selected.
2. Click the Detect button.

You can also add processors manually. See Working without your Hardware, below, for more information.

Any device in the tree can be disabled by clearing the corresponding check box. When disabled it will no longer appear in the Processing Tree in the design-time window.

To remove a device permanently:

- Right-click the desired device in the tree and click the Delete command on the shortcut menu.

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.

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

Rig Specific Device Options

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.

To configure a device or its options:

- Click the device icon in the tree diagram.
Corresponding device configuration options are displayed in the details area to the right of the tree.

- Make any desired changes.
- Click a new device in the tree or click the OK button to stop editing the rig and save your changes.

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.

Hardware specific options, such as channel count, can be configured for a particular experiment from the Processing Tree in the Synapse design-time window. See “Templates” on page 61.

**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 device 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 Check Network button.
4. In the Network Dialog box, 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.

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 design-time window, the Processing Tree is populated with the hardware in your rig. With your phantom rig you can configure your experiment.

Before running your experiment, make sure your actual hardware system matches your rig configuration before running your experiment.

**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:

The Command Bar contains the most often used elements of Synapse.

The Processing Tree displays a graphical representation of your experiment.

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.

If an input/output device does not show up automatically in the Processing Tree, you will need to edit the rig. See “The Rig” on page 43.

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.
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:

- The Processing Strip displays a drilled down look at the corresponding branch of a signal or data path.
- The Gizmos list displays tasks that can be added to the selected item.
- The Options area displays configuration options for the selected item.

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.

To display configuration options for an item in the Processing Tree:
- Click the corresponding icon in the tree diagram.

The settings are displayed in the Options area.
To delete a device or gizmo:

- Right-click the corresponding icon in the Processing Tree and click **Delete** on the shortcut menu.

The item is removed from the experiment (this does not remove a device from your rig, but you must return to the Rig Editor to make it visible again).

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 relevant gizmos are displayed in the Gizmos list. 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.

To expand or collapse a group:

- Click the ▲ or ▼ triangle/arrow to the left of the group name.

To add a gizmo to the selected item in the Processing Tree:

- Double-click the gizmo icon.

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

Gizmo Groups

**Custom**
User designed gizmos or TDT customized functions. See “Creating User Gizmos” on page 229.

**Logic**
Gizmos that perform logical tests and signals. See “Logic” on page 91.

**Neural**
Tasks associated with neural processing, including runtime visualization and controls. See “Neural” on page 99.

**Routing**
Gizmos that group, extract, or direct signals. See “Routing” on page 145.
Signal Conditioning

Tasks associated with refining or improving signals. See “Signal Conditioning” on page 161.

Stimulation

Gizmos that design, generate, and control stimuli. See “Stimulation” on page 173.

Storage

Gizmos that store various types of data. See “Storage” on page 209.

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.

Gizmo Configuration Options

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.

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. It is best not to change the store name after you have collected data with the experiment.

The Primary Source

The primary source is displayed for quick reference. The source is determined by where the gizmo is positioned in the Processing Tree or it can be set in the block diagram.

The Block Diagram Button

The BLOCK DIAGRAM button provides access to source settings and the block diagram. See the reference section for the selected gizmo for more information.
Global Options Buttons

Located below the options tabs and applied across all tabs.

Delete button  Delete the gizmo from the Processing Tree.
Revert button  Return to the last saved or ‘committed’ state.
Commit button  Save changes on all tabs.

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

In the log, information about experiment changes is organized into columns and rows. By default, changes are shown with the most recent changes at the top. You can sort the information by clicking a column header. When you find the version you need, select it and click the REVERT button.

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.
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.
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 Processing Strip

The Processing Strip can be shown as part of the Synapse main window using the Preferences dialog (see “Synapse Preferences” on page 61). 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 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 (process) to the left. In the illustration above, the number of channels at the output of the MAP1 process 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.

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.

Help Launch the software manual (PDF).

About View version number and copyright.

Exit Close the program; will prompt to save open experiments.
Categories

Click corresponding button to display.

Menu options are available in IDLE mode only, unless otherwise specified.

User (optional)

User Names
Select a user name from the list.

New
Launch a User dialog box to add a new user profile.

More
Launch the User window where you can choose an existing user or launch the User dialog box to create a new one.

Experiment

Experiment Names
Select an experiment from the list.

Undo
Return to the state before the last change.

Redo
Redo last action that was undone.

Save As
Launch the Experiment dialog box, where you save the experiment with a name, description, and icon.

Export
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

Subject Names
Select a subject name from the list.

New
Launch the Subject dialog, where you can create a new subject.

More
Launch the Subject window where you can choose an existing subject or launch the Subject dialog box to create a new subject.

Tank

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. The template is comprised of an experiment configuration file (.synexp) and the corresponding compiled object files (.rco). 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.

In this dialog you can choose to implement a required user login or change the hierarchy which determines how experiments, subjects, and data are related.

General

User Login
- Choose a user tracking method.
  - None - No user tracking.
  - Required - Users must log in with a user name.
  - Required with Password - Each user must log in with a user name and password.

Experiments
- Select check box to make experiments private per user.

Subject
- Select check box to make subjects private per user.
Synapse

Date Format
Choose a date format:
- MM/dd/yyyy
- dd/MM/yyyy
- yyyy/MM/dd

Time Format
Choose a time format:
- h:mm:ss
- hh:mm:ss

Standby Mode
Enable Standby option at runtime.

Runtime Window Layout
Save a unique window layout for each user.

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.

Data Saving

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 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.

Global Buttons

Open Preferences File
Click to open text based ini file.

OK Button
Apply changes and close dialog.

Cancel Button
Close dialog without applying changes.
This chapter covers run modes, persistence in runtime plots, 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 choice(s).

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 runtime plot. This allows users to modify parameter values before starting the experiment.

Record
- Devices are loaded and running and data is saved to the tank.

Standby
- Disabled by default and enabled in the Synapse preferences (See “Synapse Preferences” on page 61). In this mode devices are loaded and running but signals are not being acquired and saved to disk.

TDT recommends switching to idle between preview and record modes.
Runtime Plots

The plot window is automatically displayed in preview and record modes for fast, easy visualization of data.

The Data 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 Tick that displays a scalar or epoch along with subplot labeled LF1 that displays a 16 channel subplot of streamed biological data.

Subsequent 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.
Float Click and drag a tab to float and reposition the window.
Split Right-click a tab then select an option on the shortcut menu to split the main window.
Persistence
Determines the method by which settings are saved or refreshed.

Lock
Locks selected persistence. If persistence is not locked, it returns to BEST on return to idle.

Best
Use last settings of any runtime controls for current experiment and user/subject.

Last
Use the last settings of any runtime controls for the current experiment.

Fresh
Use the settings in the designtime Options area and clear any runtime settings.

User
Launch the History window.

Reset Layout
Return tab positions to default merged window configuration.

Tab Layout
Synapse also remembers information about tab layout. Information about the window layout is specific to the user and is separate from persistence information. You must have a named user for the layout to be saved. When you return to idle, you can return to the default layout using the RESET LAYOUT button on the command bar.
The Toolbar

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

The toolbar contains the following commands:

- **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**
- **Data Monitor Setup** (launch dialog)
- **Refresh**

The Shortcut Menu

Additional commands for scaling, shifting, or moving plots are available from a right-click shortcut menu (left most column).

- **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** Removes any offset that was placed on the display in the subplot window.
- **Shift Up/Down** Shifts the display up or down in the subplot window.
- **Move Up/Down** Moves the subplot up or down in the Plot Window view.
- **Make larger/smaller** Makes the available subplot area larger or smaller. The other plots are resized accordingly.
- **Hide/UnHide** Hides the selected subplot or shows a hidden subplot.
- **Show Only** Hides all subplots except the selected subplot.
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 desired subplot.

Data Monitor Setup

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

**To view data monitor settings for the plot:**

- Click the **Data Monitor** button on the plot toolbar.

**Time Span Control**

- **Span**: Set the time span (sec) of the plot window.
- **History**: Determine how much plot history will be stored for viewing purposes.

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.
- **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

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, optimized for your hardware. While Synapse provides this automation and optimization, it 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.

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.

<table>
<thead>
<tr>
<th>Signal Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Enable</td>
<td>Logic</td>
<td>Enables processes, such as data storage, through the duration of each block.</td>
</tr>
<tr>
<td>#Reset</td>
<td>Logic</td>
<td>Resets counters at the start of each block.</td>
</tr>
<tr>
<td>#iTime</td>
<td>Integer</td>
<td>TimeStamp that denotes elapsed since device synchronization (beginning of each block).</td>
</tr>
<tr>
<td>#SwFire</td>
<td>Logic</td>
<td>A pulse that triggers the onset of each sweep in sweep–based protocols (such as stimulation).</td>
</tr>
</tbody>
</table>

Time Control Waveform Diagram

[Diagram of Time Control Waveform]
RZ Options
Some options may not be available for some RZ devices.

Main Tab

Master Device Rate
Auto Check Box
Select to allow Synapse to determine the master sample rate based on the tasks assigned to the RZ.
Clear to enable the drop-down menu and set the sampling rate manually.

Load Optimization
Auto Check Box
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.

Digital I/O 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](image)

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

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.

PZ Amplifier Options

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 source that can be linked to other gizmos. All other PZs have one multi-channel floating point data source.
PZ5 Options

PZ5 Analog (left) and Digital (right) Options

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.

Name ID
Choose the name for this sub amp data source that will be visible to other gizmos.

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.
**PZ5 Options Sub-Group**

The PZ5 Options contain the logical amplifier settings for each sub amp.

**PZ5 Logical Analog Amps**

Set to Base Type

Resets the PZ5 options for the current sub amp to the default settings for the selected base type (EMG, EEG, LFP, or Single Unit).

**PZ5 Logical Digital Amps**

- **High Pass Filter**
  Select a cutoff frequency for a highpass filter that is implemented on the digital headstage.

- **Low Pass Filter**
  Select a cutoff frequency for a lowpass filter that is implemented on the digital headstage.

- **Impedance Target**
  Use to intelligently control the impedance checking circuit on the headstage.

- **Use DSP Filter check box**
  Add an additional lowpass digital filter implemented on the RZ, to remove high frequency digital noise from the incoming signal.

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

**PZ2 / PZ3/ PZ4 Options**

- **Channels**
  Type or click arrow keys to set the number of channels.

- **Fast Access**
  For a PZn 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.

**More PZ3 Options**

- **Reference Mode**
  Select the mode from the drop-down menu.

Options include the PZ2/PZ3/PZ4 options above and these additional choices.
Shared – all channels use a separate shared reference.
Individual – each even channel acts as a reference for the odd channel before it.

**Input Range**
Select the desired maximum input range.

**Map**
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.

## 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 HALS, one for sending data (UDPSend) and one for receiving data (UDPRecv).

## Adding the HAL

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 processor (such as RX or RZ devices), then click **ADD UDPSEND** or **UDP_RECV**.
3. Click **OK** to close the Rig Editor and update the Processing Tree.

## UDPSend

UDPSend handles selecting and sending data. 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](image)

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.

**UDPRcv**

![UDPRcv Block diagram](image)

UDPRcv describes the data to be received and determines how the data will be handled.

**UDPRcv Options**

![Receive from UDP Options](image)

You can choose the number of channels to be received in the UDP packet 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 enable the received packet and timing signal data source outputs, 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.
RS4 Streamer

The RS4 is a storage device and requires a multi-channel data source, which you can select from the drop-down menu in the block diagram or drag the RS4 in the Processing Tree and drop it on the source gizmo. You can also choose the range of input channels to send out the UDP port.

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 a PZ5, due to the large input range, the Float-32 format is recommended.

PHANTOM STORE generates header information in the data tank for this streaming data so that the raw files stored on the RS4 can be merged with the rest of the data and read by TDT applications.

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.

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 may want to select SEND SEV RENAME PACKETS so that the RS4 data is organized.

See the RS4 section of the System 3 Manual for more information about Data Streamer operations.
PO8e Streamer

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**

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.

See the PO8e section of the *System 3 Manual* for more information about Data Streamer operations.
RV2 Video Tracker

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

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:

- 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.

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

---

**Camera**

The Camera HAL captures images from a USB camera 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.

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.

**Note:** Frame rate is controlled by a software timer and is prone to jitter. For real-time synchronized video capture and tracking, use the RV2.

### Adding the HAL

Camera’s aren’t added to the rig automatically.

**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](image)

- **Status**
  - Shows if camera is connected or not.
- **Resolution**
  - Select a screen resolution.
- **Frame Rate**
  - Select a frame rate.
- **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](image)
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.

**IZ2 Stimulator**

![IZ2 Block Diagram](image)

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

**IZ2 Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Stimulation Mode</strong></td>
<td>Select current or voltage (IZ2/IZ2H only) mode from drop-down menu.</td>
</tr>
<tr>
<td><strong>Save Impedance to CSV</strong></td>
<td>Log impedance values into CSV file stored within the block folder, whenever an impedance check runs either manually or through the Synapse API.</td>
</tr>
<tr>
<td><strong>Compliance Monitor</strong></td>
<td>This represents the actual voltage on the output of the stimulator for the currently selected bank (see Runtime</td>
</tr>
</tbody>
</table>
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 an arrow above. 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 (μA) parameter) is presented iteratively on each channel in the currently selected bank for 500ms 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

Stop 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 Amps

RA Amp Block Diagram

RA Amp Options

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

See RZ Processors 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 is provided to allow 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 Synapse API.

Options

User Circuit

*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.
Part Four: Gizmo Reference
Logic gizmos combine logical signals originating from external hardware or other gizmos into meaningful logic states, from simple to complex, that can be stored and used by other gizmos for decision making, all in real-time.

The group includes:

- StateMaker
State Maker

State Maker provides an interface for performing logical tests on single channel inputs and combining the results into output states for storage and/or further processing.

**Key features:**
- Multi-level logical states

**Outputs:**
- Logic (multiple)

**Data Stored:**
- Epoch (x4)
  - Selectable: Full, Onset, of Offset
- Source: Value, Counter, K/M Bits, Inputs

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 are exported 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 Block Diagram](image)

**State Maker Configuration Options**

See “The Options Area” on page 26 and page 61 for more information on the gizmo name, source, global options, and displaying the block diagram.
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 are important.

Click the bit icon to toggle between possible values, that is, 0, 1, or X.

DeNoise ensures the value is stable for the specified amount of time before processing it. This is useful if the input is, for example, 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 before moving on.

**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

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. When Test is ‘True’, the mask/value is ignored and any source value greater than 0 is considered true.

Right-click on a bit icon in the Mask/Value column to change the data format from binary to decimal or hexadecimal.
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 \textit{TIME OUT (MS)} period has expired (if \textit{TIME OUT (MS)} is non-zero). If \textit{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

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.
Neural

Stimulation gizmos generate complex audio and electrical/optical stimulation, or you can import your own waveforms directly into Synapse for real-time playback. The Parameter Sequencer controls the properties of the waveforms and updates the parameters in real-time for each presentation. Stimulation playback can be fully automated, or use the runtime interface to control stimulation playback parameters manually or semi-automatically.

The Neural gizmo group includes:

- Local Field Potentials (LFP)
- PCA Spike Sorting
- Tetrode Spike Sorting
- Box Spike Sorting
- Sort Binner
- Fiber Photometry
Local Field Potentials (LFP)

The Local Field Potentials (LFP) gizmo filters multi-channel waveforms to display and record LFP activity.

Data stored:
- Stream continuous waveforms (optional)

Key features:
- Runtime control
- Filter corner frequencies (optional)

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 Plot
A multichannel streaming plot is included in the data plot tab when storage is enabled. See “Runtime Plots” on page 64 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 RUNTIME CONTROLS option is selected at designtime.
LFP Configuration Options

Filtering Tab

![Filtering Options Tab](image)

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](image)

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.
PCA Spike Sorting

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.

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

Key features:
- Threshold detection deviation from RMS
- Manual threshold available per channel
- Clustering algorithms Bayesian (default), K-Means (alternate)
- Manual cluster cutting in waveform or feature space
- Audio Monitor single channel (selectable) of analog output

PCA Spike Sorting Block Diagram

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. See “Runtime Plots” on page 64 for more information on using and customizing the plots.

PCA Spike Sorting

The runtime window includes:

- **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

| 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
Set the highpass and lowpass digital filter settings.

Thresholding
Level  Set the automatic threshold level for spike detection, in number of standard deviations from the baseline.

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
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 design-time 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 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](image)

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](image)

**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.

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

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 \((\text{window width} + 2)\) samples from when the threshold is crossed. When artifact rejection is enabled, the sort code output is delayed by an additional window width, so \((2 \times \text{window width} + 2)\) 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](image)
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.
Tetrode Spike Sorting

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 stored:
- Epoc (optional) feature state
- Epoc (optional) on pass
- Waveform (optional) on pass

Outputs:
- Filtered waveforms multi-channel float
- Sort Codes multi-channel integer

Key features:
- Threshold detection deviation from RMS
- Manual threshold available per channel
- Hunt Mode search for nearby neurons
- 2-D projections select and customize the 2-D feature space
- Manual cluster cutting in 2-D feature space
- Audio monitor single channel (selectable) of analog output

Tetrode Spike Sorting Block Diagram

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” on page 147).

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. See “Runtime Plots” on page 64 for more information on using and customizing the plot.

Tetrode Spike Sorting Tab

The runtime window includes:

- **Tool Buttons**
  Performs 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 Plots**
  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**

**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.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bypass Gate</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Both, Sort, or Hunt</strong></td>
<td>Choose to apply the settings below in sort, hunt or both modes.</td>
</tr>
<tr>
<td><strong>Pile Depth</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Projection Depth</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Clear on fill</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Wave Window Show</strong></td>
<td>Select to show the wave window.</td>
</tr>
<tr>
<td><strong>Snippet Plots Show</strong></td>
<td>Select or show pile plots in the multi-tetrode display.</td>
</tr>
</tbody>
</table>

**Prune Snippet Display**

- **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**

Set the highpass and lowpass digital filter settings.

**Thresholding**

- **Level**
  - Set the automatic threshold level for spike detection, in number of standard deviations from the baseline.
- **Polarity Negative**
  - 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 design-time interface, real-time tracking will begin on all channels, otherwise the channels will remain in manual mode.
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.

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.
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

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.

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 \( \sum (w(t) \cdot w(t) - w(t-1) \cdot w(t+1))/\text{length}(w) \), 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.

Toolbar Buttons

- 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 design-time.

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).

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:*

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.


*Keyboard Projection Plot hotkeys:*

~  
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**

See “The Options Area” on page 26 and page 61 for more information on the Gizmo name, source, global options, and displaying the block diagram.

**Sorting Tab**

Settings on this tab apply to the runtime interface and snippet storage.
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 \((\text{window width} + 2)\) samples from when the threshold is crossed. When artifact rejection is enabled, the sort code output is delayed by an additional window width, so \((2 \times \text{window width} + 2)\) 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 \((\text{0b}100001 = 33)\).

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.

Storage 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
*Monitor DAC Channel*
Select an output channel to send the monitor signal to, or set to DISABLE to turn monitoring off.
Box Spike Sorting

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 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

Key features:
- Threshold detection deviation from RMS
- Manual or auto threshold available per channel
- Sorting time voltage windows in waveform space
- Audio Monitor single channel (selectable) analog signal

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. See “Runtime Plots” on page 64 for more information on using and customizing the plots.

**Box Spike Sorting Tab**

![Box Spike Sorting Window](image)
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

**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
Set the highpass and lowpass digital filter settings.

#### Thresholding

**Level**
Set the automatic threshold level for spike detection, in number of standard deviations from the baseline.

**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 design-time 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.

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.

**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.

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**

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

**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 \((\text{window width} + 2)\) samples from when the threshold is crossed. When artifact rejection is enabled, the sort code output is delayed by an additional window width, so \((2 \times \text{window width} + 2)\) 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.
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.
Sort Binner

Sort Binner primarily serves as a sort code processor that compresses sort code data. For greater versatility it can also accept any multi-channel data.

Data Stored:
Compressed sort code counts (optional)

Key features:
User defined timing and formatting options

Outputs:
Compressed sort code counts
Multi-channel signals
Timing pulse
multi-channel integer
floating point
logic

Sort Binner is designed to work with multichannel data, and in particular, sort code outputs from spike sorting gizmos. When 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—whose outputs contain sort code data that has been compressed four channels to one—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-set intervals.

Sort Binner Block Diagram

The Main output can be sent to a UDP or other output device and can be used for closed loop control.
Sort Binner Configuration Options

See “The Options Area” on page 26 and page 61 for more information on the gizmo name, source, global options, and displaying the block diagram.

**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 (2⁴-1).
Fiber Photometry

The Fiber Photometry gizmo includes design-time and runtime control of up to four light drivers and stores and reports demodulation results using up to two sensor inputs.

Data stored:
- Stream
- Stream (optional)
- Scaler (optional)

Demodulated response signals
Broadband raw signals
Driver parameters

Outputs:
- Driver output voltage (optional)
- Demodulated signals (optional)

Up to 4 single-channel floats

Key Features:
- Runtime controls
- Runtime display
- Flexible demodulation
- Driver parameters and lowpass filter setting
- Clip indicators and response results
- Sensor x driver matrix

The Runtime Interface

Runtime Plot

A plot is added to the runtime window for visualization. See “Runtime Plots” on page 64 for more information on using and customizing the plot.
The subplots shown in a runtime plot represent data storage you chose in the design-time 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 design-time.
Runtime Controls

The runtime window includes:

**Photometry Signal(s)**

- **Lowpass Filter**
  - A knob and value entry box allows runtime control of the lowpass filter applied to the sensor input.

- **Clipping Indicator(s)**
  - Two indicators, one for each sensor, flash when the user-defined clipping threshold is approached. The clipping indicator LEDs will also light up if the input voltage is below 10uV to indicate a bad connection.

**Drivers**

- **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” on page 26 and page 61 for more information on the gizmo name, source, global options, and displaying the block diagram.
Sensor(s) Tab

Sensor A / Sensor B
- **Sensor B check box** Enables a second sensor input.
- **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 HAL 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).

Demodulator
- **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 RMS calculation. This can be modified at runtime.
Light Driver(s) Tab

Output 1 through 4

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

Frequency
Modulation frequency. Can be modified in runtime mode.

Level
Light amplitude. Can be modified in runtime mode.

Offset
The offset is tweaked to reach the linear range of the physical light driver output. Can be modified in runtime mode.

Auto Enable
When selected, lights are “on” when recording begins. Otherwise, light drivers must be turned on manually in the runtime interface.
Outputs and Data Saving Tab

Gizmo Outputs 1 through 4
Gizmo outputs are optional and may be chosen from a list of possible demodulation options (sensor x driver). Selecting an output makes the data stream available to other gizmos for further processing if desired.

Demodulator Save Options (1k Rate)
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.

Misc Saves

Store Broadband Raw Signals
When enabled, a data store containing the raw signals is generated. 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”).

The first channels of the broadband store are the raw light driver voltages, followed by the raw sensor inputs. The order would be Dv1, Dv2, Dv3, Dv4, Sensor A, Sensor B, if they are enabled.

Store Driver Parameters
All light driver parameters and timestamped and stored to disk two second after a change has been made to any of the driver parameters in the runtime user interface.
Routing gizmos provide simple ways of working with single and multi-channel signals, including combining and separating signal paths, remapping channels, and controlling signal distribution to multiple gizmos.

The Routing gizmo group includes:

- Mapper
- Injector
- Parameter Manifold
- Selector
- Merger
Mapper

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.

Key features:
- Selectable maps of common headstages, electrodes, and adapters
- Outputs: Stream remapped multi-channel waveforms

The Mapper Runtime Interface

Map Tab

At runtime the channel maps table is displayed for runtime updates, primarily to enable muting noisy channels. The map columns can be hidden to avoid accidental changes to the map.
Mapper Configuration Options

Options Tab

See “The Options Area” on page 26 and page 61 for more information on the Gizmo name, source, global options, and displaying the block diagram.

Mapper Options

Use the drop-down menu to choose an existing map for your Headstage, Adapter, or Electrode, or create your own custom map (Static). 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 “-” to delete the selected column. Click “+” 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](image)

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.
Electrode Map → Adapter Map → Headstage Map → Amplifier Map → Site Map

- Position in brain (probe positions are dependent on manufacturer).
- Remapped to adapter (if necessary).
- Remapped to headstage.
- Remapped to amplifier.
- Remapped to conventions above.

Channel numbers equal probe position in brain.

Site Numbering Conventions
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.

Data Stored:
User selected channels, when any channel changes (optional)

Key features:
Runtime manual channel control
Flexible parameter handler

Outputs:
Output  multi-channel floating point output
ChanSel-* (optional)  selected channel numbers
Par Output  parameter stream

Signal Injector Block Diagram

Injector Configuration Options
See “The Options Area” on page 26 and page 61 for more information on the Gizmo name, source, global options, and displaying the block diagram.
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
See “Parameters Table” on page 175, for more information.
Parameter Manifold

Use the Parameter Manifold if you have multiple stimulation gizmos that require parameter inputs that you want controlled from the same parameter sequencer. 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.

Data Stored:
Parameter values when triggered (optional)

Key features:
Control of multiple stimulation gizmos from a single place.

Outputs:
- Main: logic trigger
- ParOut-1..4: parameter streams to connect to stimulation gizmos
- SCout-1..4(optional): single channel, floating point parameter values

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 manifold.

5. **Configure the Parameter Manifold (see below)**.

6. Add the Parameter Sequencer to the stimulation device, such as an RZ6.

7. Connect the manifold to the sequencer.

8. Configure the Parameter Sequencer.

---

### Parameter Manifold in the Processing Tree

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

See “The Options Area” on page 26 and page 61 for more information on the Gizmo name, source, global options, and displaying the block diagram.

### 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.
You can double-click, drag, or use the Match and Reset buttons to move the unassigned parameters into master parameter rows. As you do, 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.
Parameter Control Tab

The Master Parameter Set table on the Parameter Control Tab functions like any other parameter table (see “Parameters Table” on page 175). We will control the parameters from a Parameter Sequencer gizmo, so the Mode for the parameters must be set to ParamIn.

Misc Options Tab

Use Run-time Options to show and hide run-time features.
Selector

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 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.

Data Stored:
- Selected signals (optional)
- User selected channels (optional)

Key features:
- Runtime manual parameter control
- Flexible parameter handler

Outputs:
- SigSelOut1-4 individual output channels

Selector Configuration Options

See “The Options Area” on page 26 and page 61 for more information on the gizmo name, source, global options, and displaying the block diagram.
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.

Channels

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.

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.

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 “Parameters Table” on page 175 for more information on working with parameters tables.
Merger

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.

Data Stored: None

Outputs:
- Main
- Multi-channel merged signal path

Merger Block Diagram

Merger Configuration Options

See “The Options Area” on page 26 and page 61 for more information on the gizmo name, source, global options, and displaying the block diagram.
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.
Signal Conditioning

Signal Conditioning gizmos are simple but powerful gizmos for common tasks that improve or refine input signals.

The Signal Conditioning gizmo group includes:

- Artifact Blocker
- General Purpose Filter
Artifact Blocker

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.

Data stored:
- Epoch (optional)
- Timestamps for each artifact

Key features:
- Selectable
- Timestamped logical trigger/gate values
- Onset and offset timing controls

Artifact Blocker Block Diagram

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. See “Runtime Plots” on page 64 for more information on using and customizing the main runtime data plots.
The main runtime plots show where artifact rejection has been applied to the neural signals.

**Artifact Blocker 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**

See “The Options Area” on page 26 and page 61 for more information on the gizmo name, source, global options, and displaying the block diagram.
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.

If OFFSET RELATIVE TO ONSET is selected, the gate signal timing looks like this:
General Purpose Filter

The General Purpose Filter gizmo implements highpass, lowpass, and notch filters and supports control of corner frequencies at runtime.

Data stored:
Stores None

Key features:
Runtime Controls Corner frequencies (optional)

General Purpose Filter Block Diagram

The filter gizmo applies user-defined filters to multi-channel data streams and makes this data available to other gizmos. No data is stored by this gizmo.

Configuration settings determine basic parameters, such as filter order, bandwidth, and default corner frequencies. When enabled, the corner frequency controls are added as a tabbed page in the runtime plot window.

Filter Configuration Options

See “The Options Area” on page 26 and page 61 for more information on the Gizmo name, source, global options, and displaying the block diagram.
Options Tab

Filtering Options

Enable Run Time Controls Check Box
Select or clear the check box to enable or disable runtime filter controls.

Filter settings are arranged with columns for settings and a row for each filter.

Highpass/Lowpass Filter
Filter Order Choose the number of biquad filters to use for each filter type.
Corner Frequency Type the filter frequency or drag the slider to set the initial filter values.

Notch Filters (1 - 4)
Cut Depth Select the notch depth (in dB) from the drop-down menu.
Center Frequency Type the frequency or drag the slider to set the initial frequency value.
Bandwidth Select the bandwidth of the notch in octaves.
Unary Signal Processor

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.

Data stored:
- Epoch (optional)
- Parameter values and timestamp

Outputs:
- Main: single or multi-channel floating point signal
- Parameters: varies

Unary Signal Processor Block Diagram

The Unary Signal Processor gizmo automatically detects if a single or multi-channel signal is connected. It applies the user-defined mathematical operations and makes this data available to other gizmos. The gizmo uses a bounded parameter table (see “Using Bounded Parameters” on page 175) 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. If you want to view the output data during an experiment, you can add data storage gizmo to the Unary Signal Processor gizmo’s output. See “Runtime Plots” on page 64 for more information on using and customizing the plot.

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 Bounded Parameters” on page 175, for more information.

Unary Signal Processor Configuration Options
See “The Options Area” on page 26 and page 61 for more information on the gizmo name, source, global options, and displaying the block diagram.

Options Area
The Unary Options area displays the mathematical formula along with 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.

As stages are enabled or modified the parameter table is updated to show the updated parameters in use. See “Using Bounded Parameters” on page 175 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 Types

**Biquad:** An IIR filter of a selectable shape (Lowpass, Highpass, Bandpass, Notch) and order. The \( \text{Fc-}\{N\} \) parameter determines the center frequency. If shape is ‘Bandpass’ or ‘Notch’, the \( \text{Bw-}\{N\} \) parameter determines the bandwidth in octaves. 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 \( \text{Fc-}\{N\} \) parameter determines the center frequency. The \( \text{Bw-}\{N\} \) parameter determines the bandwidth in octaves.

**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.

Operations

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 value of the ‘OpParA-3’ parameter and then converts to 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.

Working with Single and Multi-Channel Signals

The gizmo can be used with single and multichannel signals and automatically detects the number channels in the input. When the input includes multiple channels, the formula is applied independently to each channel. Each instance of the Unary Signal Processing gizmo can handle up to 96 channels. If more channels are required, you can use a second gizmo. However, splitting channels across gizmos is not recommended for 96 or fewer channels.

Tip: To pick one channel for processing from a multichannel signal, use the Selector gizmo. See “Selector” on page 157.
Stimulation gizmos are special set of tasks related to signal generation and presentation control. The Audio, Electrical, and File stimulation gizmos share a common set of bound parameters that can be defined and controlled systematically using the Parameter Sequencer gizmo. This simplifies parameter management and makes it possible to combine and control multiple stimulation gizmos with a common set of parameter values. Parameters can be held constant, be controlled by inputting values from another gizmo, or they can be controlled at runtime using a runtime slider interface.

The Stimulation gizmo group includes:

Using Bounded Parameters

- Parameter Sequencer
- Audio Stimulation
- Electrical Stimulation
- File Stimulation
Using Bounded 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.

Parameters Table

Gizmos that support bounded parameters have an Options 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.

![Parameters Tab](image)

This table allows the user to set the parameter source and bounds at design time, 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.

**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 (and is not when only saving the onset). 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 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.

If a value is entered in the Jit% column during design-time, Min and Max will be enabled. Jit%, or percentage jitter, acts as a randomizer for each presentation with Min and Max providing the bounds.

**Widget**

In Widget mode, Value and the group of adjacent parameters; Jit%, Min, and Max primarily 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 are similar to the PARAM IN line. They provide a line-in to control a parameter. The input line can come from anywhere, but must be a floating point value within the bounds defined for the parameter. You must set the parameter up in the table and commit the change, then update the source for the line-in in the block diagram.

![Block Diagram with Several Stimulus Parameter Inputs](image)

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 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 input line at any time.
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:

- On the Parameters tab, select the SHOW ALL check box and scroll to the end of the list.

After you locate the User parameters, double-click the name cell to open the Parameter Details dialog box. You must select the ACTIVE check box to enable it.
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 or Parameter Manifold gizmo to feed values to one or more parameter tables in a systematic way.

See “Parameter Sequencer” on page 181 and “Parameter Manifold” on page 153 for more information on using these gizmos.
Parameter Sequencer

The Sequencer gizmo is an interface for controlling stimulus parameters and presentation sequences. It’s a highly flexible gizmo with many options for timing, triggering, and control to cover a wide variety of stimulus presentation needs. It can be used with any of the stimulation gizmos that use a parameter table for parameter definition. By selecting the PARAM IN mode in the stimulation gizmo’s parameter table you tie that parameter to the parameter sequencer.

Outputs:

<table>
<thead>
<tr>
<th>Main</th>
<th>any</th>
</tr>
</thead>
<tbody>
<tr>
<td>StrobeOut</td>
<td>logic</td>
</tr>
<tr>
<td>SeqActive</td>
<td>logic</td>
</tr>
<tr>
<td>Current index</td>
<td>integer</td>
</tr>
</tbody>
</table>

The Parameter Sequencer inputs are highly dependent on the selections you make in the gizmo. 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.
In the example above, combinations of values for Pulse Count and Pulse Duration were saved to a file in the local working folder 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.

A slightly more useful version of the interface includes an index or sequence file. The stimuli are presented with the parameter combinations in the order listed in the 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

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” on page 26 and page 61 for more information on the gizmo name, source, global options, and displaying the block diagram.

General Tab

The General tab pulls together information about where related files will be stored and how the sequencer will be timed/triggered.

Working Directory

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 parameters are fed to the stimulus gizmo. The strobe can be based on one of three possible sources.

Manual Only

If MANUAL ONLY is selected, you must generate the strobe manually using the runtime interface. A STROBE button is provided for this method and can also be used as an additional source when you use one of the other sources.
**Strobe In**
This option enables the strobe input in the gizmo’s block diagram and requires that you select a source to be used for the strobe. Strobe Count and Continuous are also enabled to provide additional control of the strobe.

**Timer**
You can define 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.

**Working Parameter File**
In this area you must select the parameter file you will be using. Parameter files are created on the Parameter Files tab. There will be no selection available until you have created a parameter file and stored it in the Working Directory.

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 file. The **MANUAL ONLY** and **STROBE IN** options work similarly 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.

**Sequence File**
This option enables the **WORKING SEQUENCE FILE** selection where you can choose the file to control how signal presentation moves through the parameter index.

Sequence files are created on the Sequence Files tab. There will be no selection available until you have created a sequence file and stored it in the working directory.

**Parameter Files Tab**
This tab is a visual interface for selecting or creating a parameter list, to be fed to a stimulus gizmo at runtime. When the gizmo is linked to stimulus gizmo (in the Processing Tree or using the block diagram), the parameters are automatically added to the table as columns. You can BROWSE to an existing parameter file or create one. You can add rows, enter parameter values, validate the data, and save parameter files in the working directory.

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 value.

**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.

Saving the parameters file also allows you to select it in other/future Parameter Sequencer gizmos without having to recreate the table.

**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](image)

**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. 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.

When you have more than one sequence, you can choose which sequence to begin with or to lock to on the General Tab.
Audio Stimulation

The Audio Stimulation gizmo configures timing, parameter handling, and audio stimulation generation.

**Data Stored:**
- User selected parameters
- Parameter list (optional)
- Raw stimulus waveform (optional)

**Key features:**
- Runtime manual parameter control
- Flexible parameter handler
- Easy signal design and timing control

**Outputs:**
- Stimulus waveform
  - floating point
- Parameters
  - varies
- Pulse sync
  - logic
- Stim sync
  - logic

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

If enabled in the gizmo configuration, an eStim1 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.

- **Strobe Button**: Click and release to trigger a manual strobe pulse.
- **Mute Button**: Select check box to zero stimulus signal.
- **Monitor Feeds**: Select the check box to show and stimuli controlled by an Input line. 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 line.
- **Show Constant**: Select the check box to display values for parameters set to constant. They will appear gray.

Audio 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” on page 26 and page 61 for more information on the gizmo name, source, global options, and displaying the block diagram.
Waveform Tab

Timing

Duration
Choose to set the duration of the stimulus waveform: per Pulse, per Parameter, or per Strobe.

Pulse
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.

A: Stimulus Duration
B: Pulse Period
C: Pulse Duration

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.

Parameter

`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:

- set a constant value,
- use a runtime widget,
- use a parameter input from a parameter sequencer or
- use an input from a scalar input line.

See “Parameters Table” on page 175 for more on using the runtime widget or the parameter input option.

Strobe

`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.
**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.

**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, gaining 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.

**Filtering**
When a Highpass or Lowpass Active check box is selected a highpass or lowpass filter is applied. Can be used to frequency limit noise.

**Parameters Tab**

![Image of Parameters Tab]

**Audio Stim 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.

**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 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 when only saving the onset). To make your own store names, clear the Auto ID check box.

**SCount-1 and SCount-2**

Select the radio button in the desired row to feed the parameter to an output line. See “Parameters Table” on page 175 for more information on using the parameters table.

### Misc Options Tab

![Misc Options Tab](image)

**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 Options**

**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.
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 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.

Auto ID field and check box
A store name is generated automatically. To use your own store name, clear the AUTO ID check box.

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.
Electrical Stimulation

The Electrical Stimulation gizmo configures timing, parameter handling, and electrical stimulation generation.

Data Stored:
- User selected parameters
- Parameter list (optional)
- Raw stimulus waveform (optional)

Key features:
- Runtime manual parameter control
- Flexible parameter handler
- Easy signal design and timing control

Outputs:
- Stimulus waveform: floating point
- Stimulus inverse: floating point
- Parameters: varies
- Pulse sync: logic
- Stimulus sync: logic

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 output.

Electrical Stimulation Block Diagram
Electrical Stimulation Runtime Interface

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 the 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**

- **Strobe Button**: Click and release to trigger a manual strobe pulse.
- **Mute Button**: Select check box to zero stimulus signal.
- **Monitor Feeds**: Select the check box to show and stimuli controlled by an Input line. 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 line.

Show Constant Select the 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” on page 26 and page 61 for more information on the gizmo name, source, global options, and displaying the block diagram.

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 ($\times -1$) 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

**Run-Time Options**

*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.

*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 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 for 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 epoch is stored (and is not when only saving the onset). To use your own store name, clear the AUTO ID check box.

**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.

**Misc Options**

**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

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.

**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” on page 198 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.

See “Parameters Table” on page 175 for more information on using the parameters table.
**File Stimulation**

The File Stimulation gizmo plays stimulus waveforms from a list of files. It supports timing control and dynamic parameters.

**Data Stored:**
- User selected parameters
- Parameter list (optional)
- Raw stimulus waveform (optional)

**Key features:**
- Runtime manual parameter control
- Flexible parameter handler
- Easy signal design and timing control

**Outputs:**
- Stimulus waveform: floating point
- Parameters: varies
- Pulse sync: logic
- Stim sync: logic

---

**File Stimulation Block Diagram**

---

**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 the 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.

- **Strobe Button**: Click and release to trigger a manual strobe pulse.
- **Mute Button**: Select check box to zero stimulus signal.
- **Monitor Feeds**: Select the check box to show any stimuli controlled by an input line. 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 line.
- **Show Constant**: Select the check box to display values for parameters set to constant. They will appear gray.

**File 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” on page 26 and page 61 for more information on the Gizmo name, source, global options, and displaying the block diagram.

**General Tab**

![General Tab](image)
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 with 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)

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 move files.

- move a file from available to selected
- move all files to selected
- move a file from selected to available
- move all files to available
- 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.

**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 when only saving the onset). To make your own store names, clear the AUTO ID check box.

See “Parameters Table” on page 175 for more information on using the parameters table.
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.

Run-time Options

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.

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 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.
**Auto ID field and check box**

A store name is generated automatically. To use your own store name, clear the **Auto ID** check box.

**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.
Storage

Timestamp and store any type of real-time data; continuously streamed data or periodic data samples, single channel or multi-channel. All are supported.

The Storage gizmo group includes:

- Epoch Event Storage
- Stream Data Storage
- Strobed Data Storage
Epoch Event Storage

The Epoch Event Storage gizmo stores timestamps and values when triggered. Supports single channel or multiple channel input.

Data stored:
- Epoch event value (optional) and timestamp

Epoch Event Storage Block Diagram

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. See “Runtime Plots” on page 64 for more information on using and customizing the plot.

Epoch Event Configuration Options
See “The Options Area” on page 26 and page 61 for more information on the gizmo name, source, global options, and displaying the block diagram.
Trigger Options

By default, an internal timer stores timestamps at a regular interval. To use a gizmo input, clear the INTERNAL TIMER box. Then commit the change; the StrobeIn gizmo input is enabled so you can provide an external trigger source.

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.

Store Options

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 when only saving the onset).
Stream Data Storage

The Stream Data Storage gizmo is a general purpose data streaming tool that includes options for data format and scaling.

Data stored:
Stream raw or plot decimated continuous waveforms

Stream Store Block Diagram

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. This is not recommended for storing data streams.

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.

The Runtime Interface

Runtime Plot
See “Runtime Plots” on page 64 for more information on using and customizing the plot.
Stream Storage Configuration Options

See “The Options Area” on page 26 and page 61 for more information on the gizmo name, source, global options, and displaying the block diagram.

Options Tabs

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.

Choose the desired units to apply the appropriate scaling factor to the data.

Select DISCRETE FILES to save each channel of data as a discrete file (*.sev format) in the data tank.

Clear the SAVE TO DISK check box to view data in the runtime plots without storing data to the Tank.
Strobed Data Storage

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.

Data stored:
- Scaler
- Continuous
- Scalar
- Block waveforms

Strobe Storage Block Diagram

The Runtime Interface

Runtime Plot
A strobe plot is added to the runtime window for visualization. See “Runtime Plots” on page 64 for more information on using and customizing the plot.

Strobe Storage Configuration Options
See “The Options Area” on page 26 and page 61 for more information on the gizmo name, source, global options, and displaying the block diagram.
Options Tab

Capture Mode

The Timing and Storage Options have different meaning depending on which capture mode is selected.

Single Scalar

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.

Fixed Duration

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.

If the rate is below 10 Hz, each stored data point will have a timestamp and value (Record Type Scalar).

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.

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.
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.

**Strobe Controlled**

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](image1)

![Strobe Controlled Block Timing Diagram](image2)

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.

**Continuous**

In Continuous mode, the Strobe input is ignored and data is continuously recorded into the tank at the specified sampling rate. For high sampling rates of continuous data above 30Hz, the Stream Data Storage gizmo is recommended instead.
Visualization

View real-time data.

The Visualization gizmo group includes:

oScope
Oscilloscope

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.

Data stored:
- Epoch (optional) feature state
- Epoch (optional) on all conditions met
- Strobe waveform (optional) on all conditions met

Key Features:
- Runtime control timing and threshold
- Threshold detection manual and auto

Outputs:
- Feature state logic
- Trigger (on all conditions met) logic
- Strobe (on all conditions met) logic
- DelayedSig float

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.

Oscilloscope Block Diagram
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. See “Runtime Plots” on page 64 for more information on using and customizing the plot.

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.

Oscilloscope Configuration Options
See “The Options Area” on page 26 and page 61 for more information on the gizmo name, source, global options, and displaying the block diagram.

In edit mode, the oscilloscope gizmo displays a simulated waveform alongside the gizmo options. This interface expedites setting the feature and triggering options.

Channels
Up to four channels can be input. 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.

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Above</td>
<td>signal is above the threshold</td>
</tr>
<tr>
<td>Below</td>
<td>signal is below the threshold</td>
</tr>
<tr>
<td>Between</td>
<td>signal is between an upper and lower threshold (when selected a second threshold marker is added to the plot)</td>
</tr>
<tr>
<td>Outside</td>
<td>signal is outside an upper and lower threshold (when selected a second threshold marker is added to the plot)</td>
</tr>
<tr>
<td>Rising</td>
<td>signal is rising</td>
</tr>
<tr>
<td>Falling</td>
<td>signal is falling</td>
</tr>
</tbody>
</table>

Display Sync
The Display Sync determines how to align the waveform at X=0.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature</td>
<td>when selected feature is true</td>
</tr>
<tr>
<td>Ext Input</td>
<td>external input from another gizmo or digital input (defined in the block diagram)</td>
</tr>
<tr>
<td>None</td>
<td>not set, shows the free running traces (ambient traces)</td>
</tr>
</tbody>
</table>

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 above, 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 a 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**

**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**

The **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**

The 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: store the Feature state on/off timestamps. Feature state remains true/high (1) as long as the Feature condition is met.
On Pass: store 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: Strobe starts when Trigger fires (all conditions are met) and remains high for the duration of the X-axis window (range).
DelayedSig: The DelayedSig output 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.
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.

MultiChannel 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

- Move Offsets: click-and-drag
- Y-axis Zoom: mouse-wheel
- X-axis Zoom: Ctrl + mouse-wheel
Part | User Gizmos
Five:
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, or by clicking the icon 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’.

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 ‘Main’, ‘Alt-1’, ‘Alt-2’, ‘Alt-3’. If the user gizmo requires at least one data source, you do not need a ‘Root’ gizmoInput macro in your circuit.

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 ‘Main’, ‘Alt-1’, ‘Alt-2’, ‘Alt-3’. Each output requires its own gizmoOutput macro, where you specify the name, data format and allowed channel count range.

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](image)

**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 use 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 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 Synapse API 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

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Graphical Representation</th>
<th>Synapse Alternative</th>
</tr>
</thead>
</table>
| **zHop Components**
| [Image](#) | Use the gizmoInput / gizmoOutput macros to share signals between gizmos. |
| MCzHopOut, MCzHopIn | [Image](#) | The standard timing zHopIns from OpenEx (ITime, Reset and Enable) can be used in the circuit. |
| **Pipe Components**
| PipeSource | [Image](#) | Use the gizmoInput macros to share signals between gizmos. |
| PipeOut, PipeIn | [Image](#) | |
| MCPipeOut, MCPipeIn | [Image](#) | |
| **DSP Assign**
| DSP: 1 | [Image](#) | Use DSP assignment option in gizmoInput macro to force a user gizmo to run on a specific DSP. |
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.

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Graphical Representation</th>
<th>Synapse Alternative</th>
</tr>
</thead>
<tbody>
<tr>
<td>SourceFile</td>
<td>![SourceFile Diagram]</td>
<td>Use Synapse API to load a buffer instead.</td>
</tr>
<tr>
<td>ShortDynDelay, LongDynDelay</td>
<td>![ShortDynDelay Diagram] ![LongDynDelay Diagram]</td>
<td>None</td>
</tr>
<tr>
<td>ReadBuf, WriteBuf</td>
<td>![ReadBuf Diagram] ![WriteBuf Diagram]</td>
<td>None</td>
</tr>
</tbody>
</table>

Unsupported RPvdsEx Components

<table>
<thead>
<tr>
<th>Component Name</th>
<th>Graphical Representation</th>
<th>Synapse Alternative</th>
</tr>
</thead>
<tbody>
<tr>
<td>SourceFile</td>
<td>![SourceFile Diagram]</td>
<td>Use Synapse API to load a buffer instead.</td>
</tr>
<tr>
<td>ShortDynDelay, LongDynDelay</td>
<td>![ShortDynDelay Diagram] ![LongDynDelay Diagram]</td>
<td>None</td>
</tr>
<tr>
<td>ReadBuf, WriteBuf</td>
<td>![ReadBuf Diagram] ![WriteBuf Diagram]</td>
<td>None</td>
</tr>
</tbody>
</table>

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.
Creating User Gizmos
Index

A
analog input 73
analog output 74

B
block diagram 27

C
Camera HAL 82
Connections diagram 58

D
data tank 5, 10, 22, 29
design time 23, 51
digital I/O 72
Disabled devices 45

E
Enabling Devices 14
Experiment Selection Window 18, 58
Experiment Selection window 16

F
Fiber Photometry 139

G
general purpose filter 167
gizmos 2, 17, 23, 55, 167
artifact blocker 163
audio stimulation 187
box spike sorting 129
electrical stimulation 195
epoch event storage 211, 221
fiber photometry 139
file stimulation 201
injector 151
LFP 101
mapper 147
merger 159
parameter manifold 153
parameter sequencer 181
PCA spike sorting 105
selector 157
sort binner 137
state maker 93
stream data storage 213
strob data storage 139, 215
tetrode spike sorting 115
user gizmos 3, 229

H
Hardware Abstraction Layer (HAL) 1, 159
History Window 21, 30

I
IZ2 stimulator 84

M
Medusa Amplifiers 86

O
OpenEx 5, 31

P
parameters 153, 155, 175, 184, 200, 205
parameters table 156, 175, 182, 192, 200, 205
persistence 4, 20, 37, 65
Preferences 30, 34, 61
Processing Strip 59
processing tree 24, 52
PZ amplifier 15, 46, 52, 74

R
RA amplifiers 86
Revision Log 57
rig 13, 43
Rig Hardware Abstraction Layer (HAL) 16
RS4 streamer 79, 80
runtime 20
runtime modes 63
runtime plot 20, 64
RV2 video tracker 81
RX processors 87
RZ Processors 72

T
templates 1, 16, 61
triggers 71, 165

U
UDP 77
UDPRcv 78
UDPSend 77
user gizmos 3, 229

W
Warning 47
widgets 176
window layout 22, 65