

# SuperLab<sup>®</sup> 6 Manual

## Copyright Information

© Copyright 1990-2020 Cedrus Corporation. All Rights Reserved.

Under the copyright laws, this manual or the software described within may not be copied, in whole or in part, without the written consent of the manufacturer, except in normal use of the software to make a backup copy. The same proprietary and copyright notices must be affixed to any permitted copies as were affixed to the original. This exception does not allow copies to be made for others, whether or not sold, but all of the material purchased (with all backup copies) may be sold, given, or loaned to another person. Under the law, copying includes translating into another language or format. The information contained in this publication is subject to change without notice. Cedrus Corporation makes no warranty of any kind with regard to this written material and shall not be held liable for errors contained herein or for incidental or consequential damages that may occur in connection with the use of this manual or the software, even if advised of the possibility of such damages. See also the License Agreement at the end of the manual. Cedrus and SuperLab are registered trademarks of Cedrus Corporation. All other products and brand names are trademarks of their respective holders.

**Cedrus Corporation**  
P.O. Box 6309  
San Pedro, CA 90734  
United States

<b>Other:</b>	+1-310-548-9595
<b>Web:</b>	<a href="http://www.cedrus.com">www.cedrus.com</a>
<b>Email:</b>	<a href="mailto:cedrus.com/send_email.htm">cedrus.com/send_email.htm</a>

# Hello

SuperLab is a tool for building experiments that we hope will serve you well. As of this writing, it is in its 28th year and has been sold in over 70 countries.

This would not have been possible without you, our dear users. From the bottom of our hearts: **thank you**.

If you like SuperLab, please tell your colleagues about it. This would be helpful to us.

Thanks again for choosing SuperLab. We wish you success in your work.

# TABLE OF CONTENTS

		Page
Chapter 1	<b>Introduction</b> Installation & System Requirements ▪ Licensing ▪ Support	1
Chapter 2	<b>Getting Started with SuperLab</b> A half hour tutorial that's worth every minute	3
Chapter 3	<b>More In-Depth Features</b> Determining the Correct Response ▪ Providing Feedback ▪ Using Tags ▪ Sending Event Codes to EEG/ERP	17
Chapter 4	<b>Parameters</b> Introduction ▪ A Taste of Parameters ▪ Parameter Types ▪ Built-in Parameters	23
Chapter 5	<b>Conditional Branching</b> Living Without Rules ▪ A Memory Experiment ▪ Rules of Rules	29
Chapter 6	<b>Randomization</b> Randomizing The Order in Which Trials Are Presented ▪ Sublists ▪ Varying an Aspect of a Presented Trial ▪ Randomizing Per Group	39
Chapter 7	<b>Running Experiments &amp; Viewing Data</b> Things to Know Before You Run Experiments ▪ Data Viewer	49
Appendix	<b>License Agreement</b>	59

# Also From Cedrus

## SUPERLAB REMOTE

Allows participants to download your experiment and run it on their local Mac or PC.

## SUPERLAB X6

SuperLab X6 provides “live” integration with Tobii and SMI eye trackers, allowing for gaze contingent experiments.

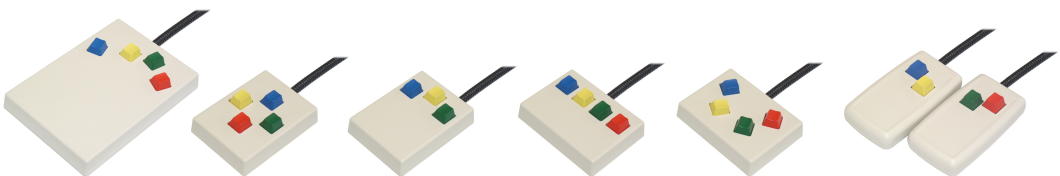
## RB SERIES RESPONSE PADS

Available in four layouts, each pad ships complete with white, red, green, blue, yellow, and clear key tops to suit different experiments. Add an m-pod to send event markers.



## LUMINA fMRI RESPONSE PADS

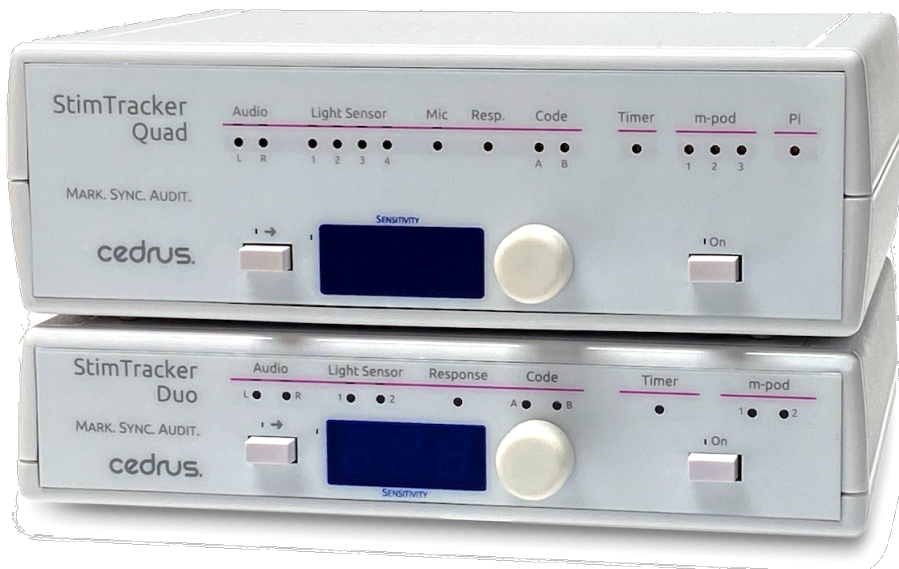
These reliable fMRI pads are built using plastic and fiber optics and support scanner trigger input so you can synchronize the start of the trials.





## STIMTRACKER

This is a true plug-and-play solution for sending event markers and onset of stimuli data to one or multiple physiological recorder, e.g. EEG, NIRS, or eye tracker. StimTracker can autonomously detect the onset of visual stimuli, auditory stimuli, and participant key press or vocal responses.



*“Why is our memory good enough to recall to the last detail things that have happened to us, yet not good enough to recall how often we have told them to the same person.”*

*- François de La Rochefoucauld,  
French writer, (1613-1680).*

# Introduction

Thank you for purchasing SuperLab, your tool for building experiments. With SuperLab, you can present images, text, and video, and collect responses accurately from participants.



**At a minimum**, we strongly recommend that you go through the tutorial chapter 2, [Getting Started with SuperLab](#), and read the short but important chapter 7, [Running Experiments & Viewing Data](#).

## INSTALLATION & SYSTEM REQUIREMENTS

On the **Mac**, copy the SuperLab and Cedrus Data Viewer applications to your Applications folder. Mac OS 10.12 or later is required.

On **Windows**, run the installer that you have downloaded. Installation requires Windows 7, 8, or 10. For using movies, you need DirectX 9 and a video card that works with DirectX 9, including support for hardware acceleration and VMR.

SuperLab can play movies on a single core or single processor computer, but playing movies smoothly requires a dual core computer. We recommend at least 4 GB for memory. Having a very good video card is more important than the processing speed of the CPU.

## LICENSING

### Activating Your License

When you buy a SuperLab license, you receive a license key. When you run the software for the first time, it will prompt you to create an account. Once created, you will no longer need the license key; all future licensing-related actions will need only your email address and password that you used when creating the account.

A SuperLab license cannot be deactivated. Instead, SuperLab can be activated on a computer for up to one year, and can be reactivated as many times as you like. This approach offers a lot of flexibility, e.g. you can install SuperLab on a computer at a university that you are visiting and activate it only for the duration of your visit.

### Issuing Leases

You can get information about your licenses from within SuperLab itself by clicking on the **Help** menu and then **My License Activity...** This is particularly useful if you own several “seats”.



If you have two or more seats, you can take advantage of SuperLab's unique [leasing feature](#). It is like a library checkout system. You can go to the **Help** menu and choose **Create Leases....** This feature allows you to issue a lease for, say, two months. When the lease expires after two months, that license is automatically returned and available to you again.

## CEDRUS SUPPORT

### Via Community Forums

The forums are the best way to start, first by searching existing posts to see if your question has been already been asked and answered. You can always post your own question.

[community.cedrus.com](https://community.cedrus.com)

### By Email

Due to spam, we no longer publish an email address; support emails can be sent via:

[cedrus.com/send\\_email.htm](https://cedrus.com/send_email.htm)

### By Phone

The hours are 9 AM to 4 PM California time. The number is +1-310-548-9595.

# Getting Started with SuperLab

In this tutorial, we will build a word/non-word experiment that will present instructions to the participant followed by six trials. Each trial will consist of a cue and stimulus.

When developing your own experiments, it is important to plan carefully and take advantage of the stimulus lists and tags features. This can save you a lot of development time, will make it easier to see the structure of your experiment and maintain it, and will make analyzing the data easier later on.

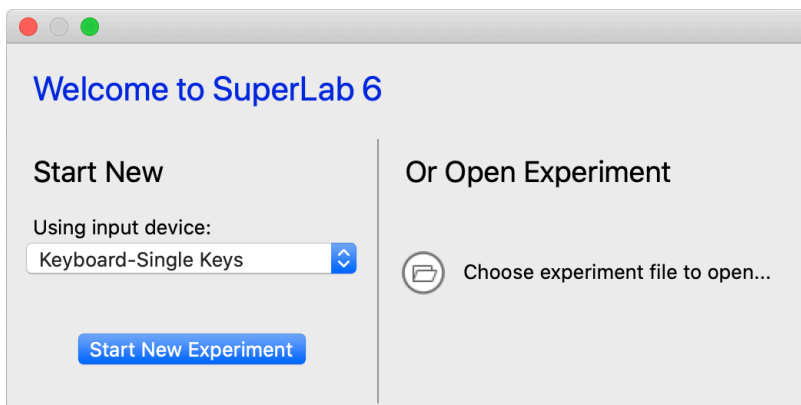


It bears emphasizing: **you should use stimulus lists** in your experiment whenever possible. This will save you time, allow you to add new items to the lists without affecting the experimental design, and provide you with more options, e.g. two-level randomization.

Going through this tutorial takes about half an hour. What you need to type is shown in **this color**.

## PARTICIPANT INPUT

Most experiments require a response from the participant. When you first open SuperLab, it prompts you to select an input device:



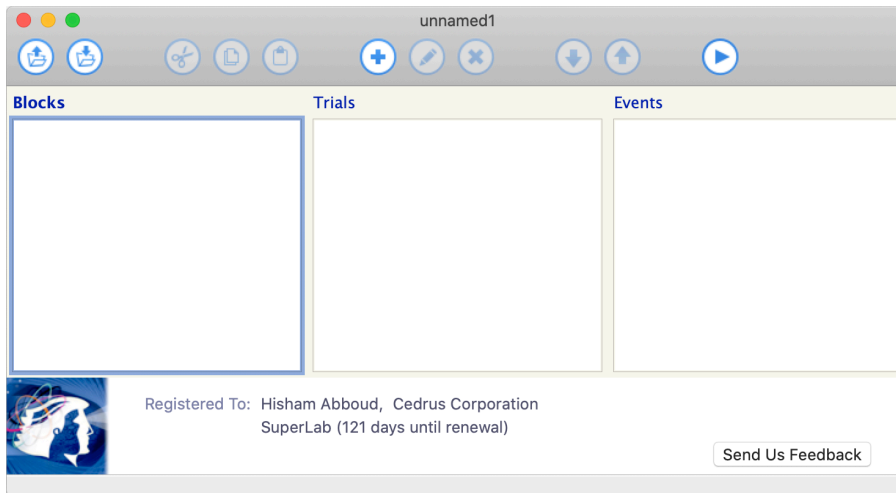
We will use the keyboard for this experiment. There are two keyboard options. The first, single key input, is used when the response is expected to be a single key press or key release, e.g. Y

or N. The second, string input, is used when you want the participant to type words or sentences.

- Under Start New on the left, select “Keyboard-Single Keys”
- Click on the **Start New Experiment** button

## CREATING A BLOCK AND A TRIAL

The smallest unit of an experiment is the event, typically a visual or auditory stimulus. A trial is a set of events and a block is a set of trials. To run an experiment, you must have at least one block, one trial, and one event that are linked together.



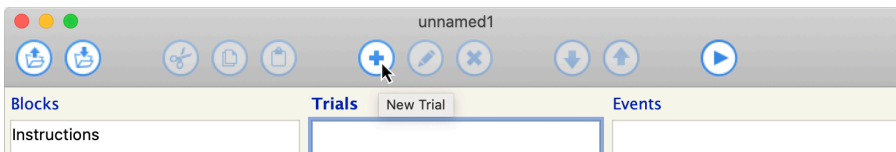
Note the blue rectangle drawn around the list of blocks. To familiarize yourself with the SuperLab workspace, try the following steps:

- Click on the **Edit** menu. A menu will appear.
- Choose the menu item **One New Block**. This will cause a dialog (screen) to appear. This is the Block Editor.
- Type “**Instructions**”.
- Click on the **OK** button. The Block Editor will close and “Instructions” will appear in the list of blocks

Now we will create the first trial:

- In the main experiment window, click on the list of trials, currently empty. The blue rectangle will move from the list of blocks to the list of trials.
- Click on the **Edit** menu again.

- Note how some menu items have changed. The one that used to read **One New Block** is now **One New Trial**. The **Edit** menu is dynamic; some of its items change depending on your actions.
- In the toolbar, click on the **New Trial** icon shown below. The Trial Editor will appear.



Notice how SuperLab displays yellow “tooltips” as you move the mouse cursor over the icons.

- Type “**Instructions**”.
- Click on the **OK** button. The Trial Editor will close and “Instructions” will appear in the list of trials.

You can also press the Tab key to move the blue selection rectangle.

## SAVING THE EXPERIMENT

We may not have created much of an experiment so far but it’s a good idea to save, and save often:

- Click on the **Save Experiment** icon (floppy disk) in the toolbar. You can also click on the **File** menu and select **Save**.
- SuperLab will ask you to provide a name for your experiment. The default filename extension is “SL6”. For example, if you type “**tutorial**”, SuperLab will save a file as tutorial.sl6.



**Important:** Do not save the experiment to a server. Nor should stimuli files be located on a server. This will introduce delays, sometimes considerable, that are beyond your control. Double-check that your Desktop and Documents folders are not mapped to a server.

## CREATING AN EVENT

Back to the experiment, you are now ready to create your first event. We want to complete the instructions part of the experiment.

- In the main experiment window, click on the list of events. The selection rectangle will move from the list of trials to the list of events.
- In the toolbar, click on the New Event icon; the Event Editor will appear.

As mentioned earlier, an event is the smallest unit of an experiment. It is where you tell SuperLab what to present to the participant, what is a correct response, and how to provide feedback based on the participant's response.

- For the event name, type “**Instructions**”.
- Click on the **Stimulus** tab to select it, if it's not selected already.
- Click on the Event Type popup menu and select Text.
- In the large text editing field, type:

You are about to be presented with some words.

Press the letter Y if you recognize the word or  
the letter N if you do not.

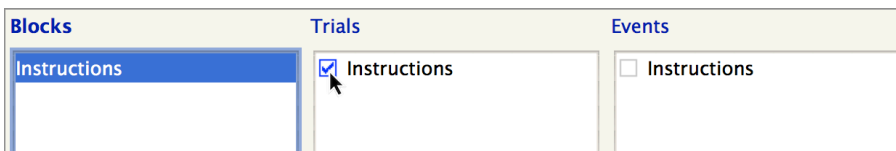
Press any key to start the experiment.

Click on the **OK** button. The Event Editor will go away and “Instructions” will appear in the list of events.

## LINKING

In SuperLab, various elements of an experiment are created separately and then “linked” together. To link the block, trial, and event that you just created:

- Click (only once) on the block “Instructions” to select it
- Click on the blue checkbox that's on left of the trial named “Instructions”




The block and trial are now linked. Let's do the same for the trial and event:

- Click on the trial “Instructions” to select it (the trial name itself and not its checkbox).
- Click on the blue checkbox that's on the left of the event name.

Click on the **Save Experiment** icon .

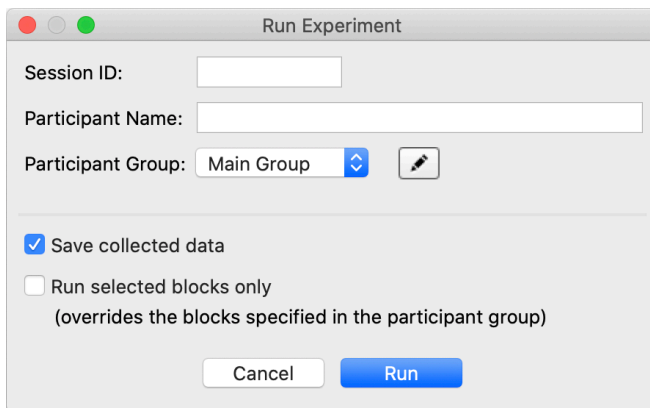
## LET'S RUN

You now have enough to run an experiment. Click on the **Run Experiment** icon  on the toolbar. SuperLab will present the Run dialog, shown on the next page.

Since we're still early in the development stage, there is no need to enter the participant's name or collect data:



- Click on the **Save collected data** checkbox to turn it off
- Click on the **Run** button

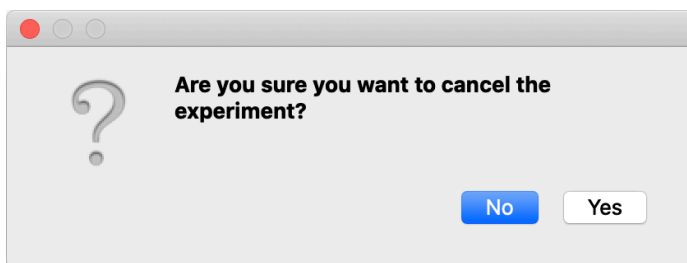


The 'Run Experiment' dialog box contains the following elements:

- Session ID:** A text input field.
- Participant Name:** A text input field.
- Participant Group:** A dropdown menu currently showing 'Main Group' with a blue arrow icon, and a pencil icon for editing.
- Save collected data:** A checked checkbox.
- Run selected blocks only:** An unchecked checkbox, with a note below it: '(overrides the blocks specified in the participant group)'.
- Buttons:** 'Cancel' and 'Run' buttons at the bottom.

SuperLab will open a window that covers the entire screen, display the instructions that you typed earlier, and wait for you to press any key on the keyboard.

You can press the **Esc** key to cancel an experiment in progress, at which point SuperLab will prompt you to confirm:



The confirmation dialog box features a large question mark icon and the text: 'Are you sure you want to cancel the experiment?'. At the bottom right, there are two buttons: 'No' (highlighted in blue) and 'Yes'.



You can take advantage of this feature to **pause** an experiment instead of stopping it by clicking on the **No** button. SuperLab will then repeat the trial that was interrupted and continue with the experiment.

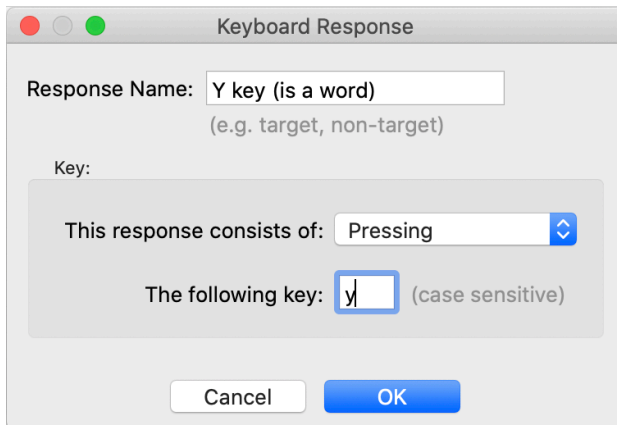
## DEFINING RESPONSES

As mentioned in the instructions, the participant is expected to respond by pressing the Y and N keys. We need to create two corresponding responses. SuperLab will always record what the user presses. However, by explicitly telling it what is the correct response:

- You can choose to end an event only after a correct response instead of any response
- You can provide feedback to participants based on whether the response is correct or not
- The responses will be automatically coded for correctness in the collected data file (see [Determining the Correct Response](#) in the next chapter)

To create the responses:


- Click on the **Experiment** menu and select **Participant Input**. A dialog opens. The “Keyboard-Single Keys” input device is already enabled (checked) and selected.
- Click on the **Responses** tab if it’s not selected already.
- Click on the **New...** button to create a response. Another smaller dialog opens:



- For the response name, type “**Y** key (is a word)”. It is a good idea to type a descriptive name of the key instead of simply “Y”.
- Type “**y**” for the actual key.
- Click on **OK** to close the response editor.

We repeat the steps above for the N key:


- Click on the **New...** button to create a response. The response editor opens.
- For the response name, type “**N** key (is not a word)”.
- Type “**n**” for the actual key.
- Click on **OK** to close the response editor.
- Click on **Done** to close the Participant Input dialog.

Remember to save; click on the **Save Experiment** icon .

## ADDING A CUE

We are now ready to build the core of the experiment. Recall that we want to present six trials, each consisting of a cue and a stimulus. Let’s start with the cue:

- Press **Tab** until the blue selection rectangle surrounds the list of events.
- We want the new cue event to appear after the instructions, so make sure that the existing event called “Instructions” is not selected. You can do this by clicking anywhere on the white space under the event name, or by choosing **Select → None** from the **Edit** menu.

- When running an experiment, for any given trial, SuperLab presents events in the order in which they appear in the main experiment window.
- Click on the **New Event** icon  on the toolbar; the Event Editor appears.
- For the event name, type “**Cue**”.
- Click on the **Stimulus** tab if it’s not selected already.
- The Event Type should still be set to Text.
- In the large text editing field, type “+”.

A cue is typically presented for a brief moment to direct the participant’s attention to a particular location on the screen:

- Click on the **Input** tab to select it.
- Click on the **...or a time limit, whichever happens first** checkbox to turn it on.
- In the edit field below the checkbox, type “500” to have the cue displayed for half a second.
- Click on the **After any response from the participant** checkbox to turn it off.
- A participant response during a cue is typically ignored. To reduce the clutter in the collected data file, click on the **Record and save response** checkbox to turn it off.

We’re done with the cue. Click on **OK** to close the Event Editor.

## STIMULUS LISTS

The next step is to create the events that present the stimuli. To avoid needing to create six events, one for each of the six words, we introduce the stimulus lists feature. This allows you to create a single event that results in six presentations:

- Click on the **Experiment** menu and select **Stimulus Lists**. A dialog opens.
- Click on the “+” icon. A drop-down menu appears; select **Text**.
- For the list name, type “**My Word / Non-Word list**”.
- In the large edit field, enter three words and three non-words by typing:

apple

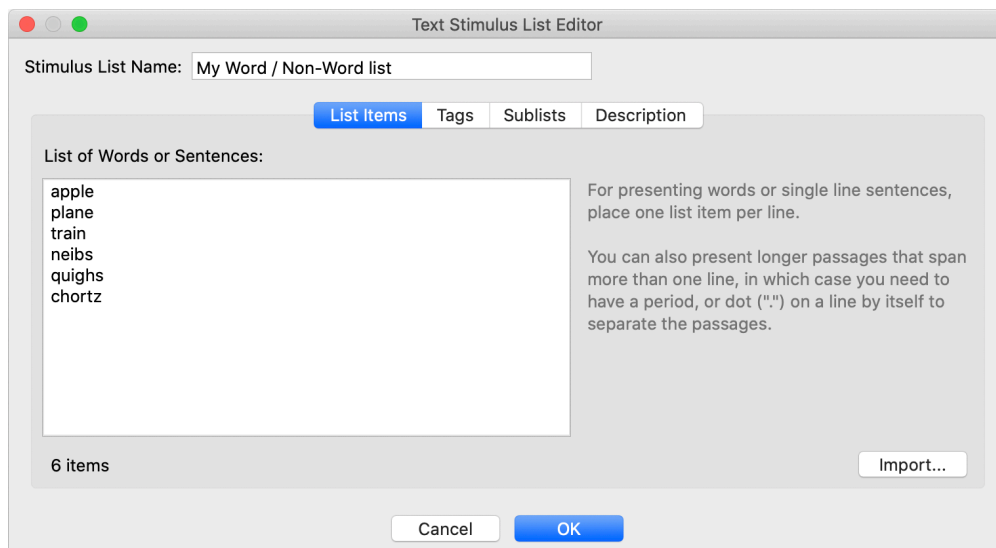
plane

train

neibs

quighs

chortz



- Click on **OK** to close the Stimulus List Editor.
- Click on **Done** to close the Stimulus Lists dialog.


The list is ready. Before we proceed to using it, it's worth taking a minute to emphasize the advantages of using stimulus lists:

- You can create a single event with many items. This becomes more and more important as the experiment gets larger.
- When you have a few events that use stimulus lists, it is easier to see the structure of the experiment in the main window.
- It is easier to make changes to the experiment.
- You can increase the size of the experiment by simply modifying the list and without touching the structure of the experiment. Trying to do so without using stimulus lists is more error prone and costly.

Click on the **Save Experiment** icon .

## CREATING/IMPORTING STIMULI

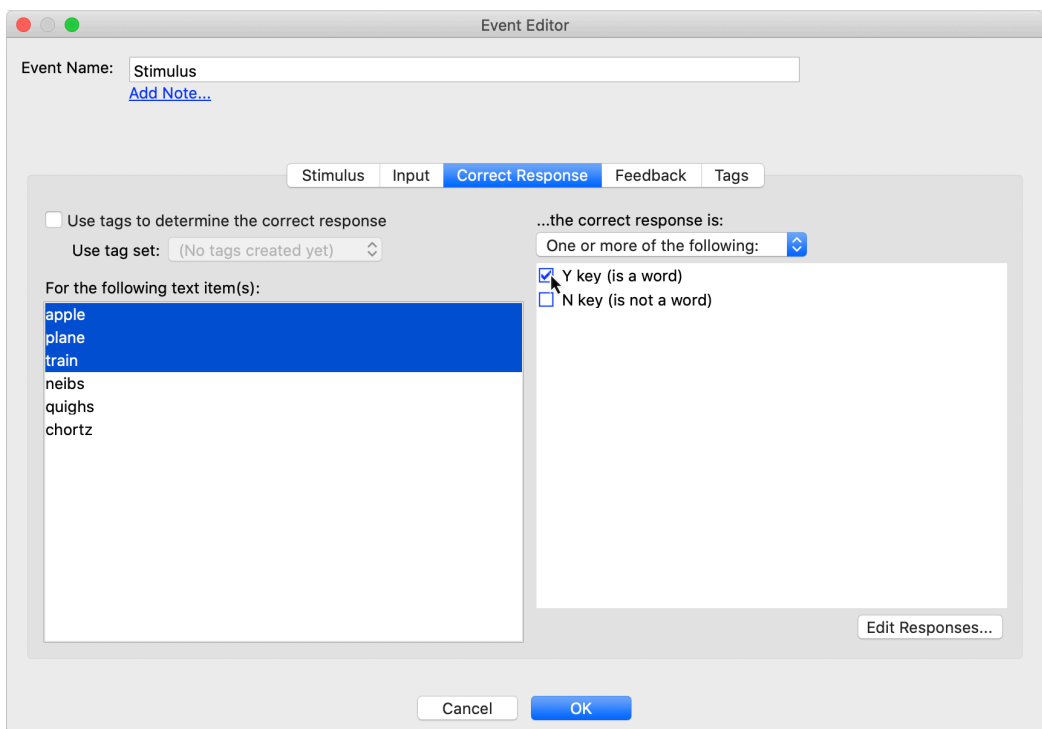
You now have everything needed to create the stimuli using a **single event**:

- Press **Tab** until the blue selection rectangle surrounds the list of events.
- Make sure that none of the existing events are selected. You can do this by clicking anywhere on the white space under the events.
- Click on the **New Event** icon  on the toolbar. The Event Editor appears.
- For the event name, type "**Stimulus**".

- Click on the **Stimulus** tab if it is not selected already.
- The Event Type should still be set to **Text**.
- Click on the Use Text option and select **Text Chosen From the List**. The name of the list that you created in the previous step appears.

That's it. You have just taken care of presenting your stimuli. We still need to tell SuperLab what is the correct response for a given list item.

- With the Event Editor still open, click on the **Correct Response** tab. The items in your stimulus list are visible in this tab.
- Select the first three words: apple, train, and plane. You can simultaneously select all three by holding down the Ctrl key on Windows or the Command (⌘) key on Mac while you click with the mouse.



- Click on the “...the correct response is” popup menu (in the upper right part of the dialog) and select the **One or more of the following** option.
- Click on the checkbox to the left of “Y key (is a word)” to turn it on.

We’re done with the list of words. We repeat the steps for non-words:

- Select the three nonwords: neibs, quighs, and chortz.

- Click on the “...the correct response is” popup menu and select the **One or more of the following** option.
- Click on the checkbox to the left of “N key (is not a word)” to turn it on.

We would like the event to stay on the screen until participant presses the correct key:

- Click on the **Input** tab.
- Click on the **After a correct response from the participant** checkbox to turn it on.


Last but not least (as far as this event is concerned), we need to reset the reaction time timer. When a trial contains more than just one event, pay particular attention to the **Reset RT Timer** checkbox.

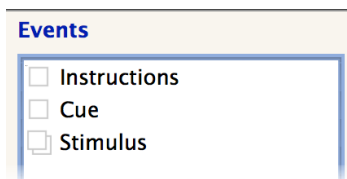



This is an **important detail**: not doing so means that the RTs collected by SuperLab will be measured relative to the onset of the cue instead of the onset of the stimulus itself. This is discussed in more detail in the section [Resetting the RT Timer](#) in Chapter 7.

If none of your events have this option enabled, the reaction time will be measured relative to the onset of the first event in the trial because SuperLab has no way of knowing what is a cue and what is an actual stimulus:


- Still in the **Input** tab, click on the **Reset RT timer** checkbox to turn it on.
- Click on **OK** to close the Event Editor.

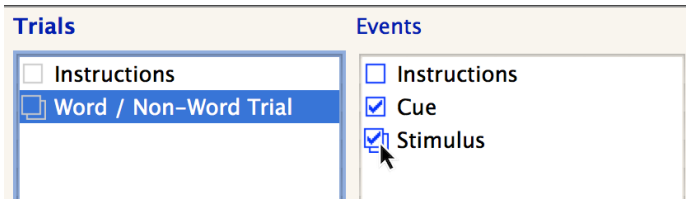
Back in the main window, notice that the Stimulus event has a stacked checkbox  instead of a standard checkbox. This is a visual indication that the event uses a stimulus list.



Remember to save; click on the **Save Experiment** icon .

Things get easier from here. We still need to create one trial and one block, and link them together. We start with the trial:


- Press **Tab** until the blue selection rectangle surrounds the list of trials.
- Click on the **New Trial** icon  in the toolbar. The Trial Editor appears.
- Type “**Word / Non-Word Trial**”.
- Click on the **OK** button. The Trial Editor will go away and “Word / Non-Word Trial” will appear in the list of trials. Click on it to select it.

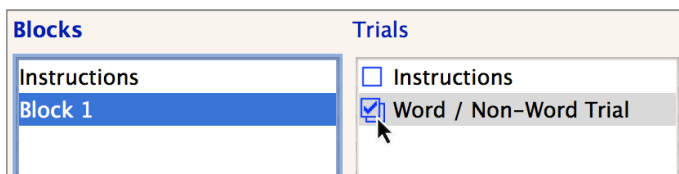


- Click on the checkboxes to the left of the events “Cue” and “Stimulus”. This tells SuperLab that trial “Word / Non-Word Trial” consists of event “Cue” followed 500 milliseconds later by the stimulus.

## CREATING THE BLOCKS

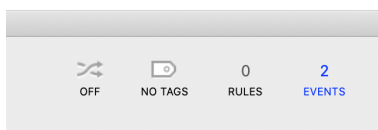
It would be a bad idea to link the trial “Word / Non-Word Trial” to the existing “Instructions” block. Not only is it not a good practice, but in the case of this experiment it is not an option because we want to randomize the trials. If everything was part of a single block, the instructions would get randomized as well.

- Press **Tab** until the blue selection rectangle surrounds the list of blocks.
- Click on the **New Block**  icon in the toolbar. The Block Editor appears.
- Type “Block 1”.
- Click on the **OK** button. The Block Editor will close and “Block 1” will appear in the list of blocks. Click on it to select it.
- Click on the checkbox to the left of the trial “Word / Non-Word Trial” to link it.



## THE AT-A-GLANCE ICONS


Double-click on the trial “Word / Non-Word Trial” to edit it. When the Trial Editor appears, notice the four small icons in the top right:

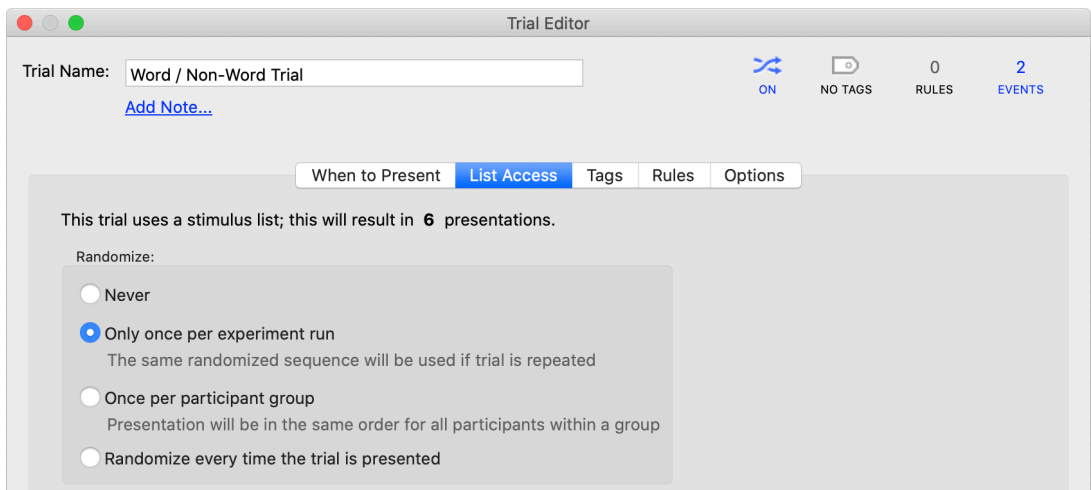


These icons provide some information about the trial. From left to right, they indicate if randomization is turned on, if the trial is tagged, the number of rules used by the trial, and the number of events that the trial uses (two in this case).

## RANDOMIZING


If the Trial Editor is not open, double-click on trial “Word / Non-Word Trial” to edit it, and:

- Click on the **List Access** tab
- Click on the **Only once per experiment run** randomization option; the randomization  at-a-glance icon will change color
- Click the **OK** button to close the Trial Editor



Click on the **Save Experiment** icon .

Randomization is a topic that is worthy of its own chapter; see [Chapter 6](#).

Congratulations! You have learned several concepts of SuperLab such as linking and using stimulus lists. You now have a ready-to-go experiment. Click on the **Run Experiment** icon  on the toolbar to run it.

The rest of this chapter provides a quick overview of the block and trial editors.

## A CLOSER LOOK AT THE BLOCK EDITOR AND TRIAL EDITOR

### Conditional Branching

At some point, you may want to build more advanced experiments that involve presenting a trial or block conditionally, or perform certain actions such as incrementing counters. The Block Editor and Trial Editor have two tabs: **When to Present** and **Rules**:





Both tabs work consistently in both editors. In the **When to Present** tab, you can specify criteria that determines whether the block or trial should be presented.

In the **Rules** tab, you can specify actions that are performed if certain criteria are met. See chapter 5, [Conditional Branching](#), for more details.

## Extra Actions

There are some tasks that are common to a lot of experiments, e.g. giving the participant a break after, say, every 50 trials. The **Extra Actions** tab in the Block Editor simplifies this by offering options that reduce or eliminate the need to create rules:

The screenshot shows the 'Extra Actions' tab selected in the Block Editor. The tab bar at the top includes 'When to Present', 'Randomize', 'Looping', 'Extra Actions' (highlighted in blue), and 'Rules'. The main content area has three sections, each with a checkbox and a description:

- ☐ Stop presenting trials past  seconds. SuperLab will not interrupt a trial in progress, but will stop presenting new trials after the time period has elapsed.
- ☐ Present a maximum of  trials. This count does not include trials that may be presented again due to feedback.
- ☐ Pause experiment every  trials with message:  

Please take a break.  
Press any key when you are ready to continue.

## Trial Options

The Trial Editor's **Options** tab provides finer control over the running of an experiment:

The screenshot shows the 'Options' tab selected in the Trial Editor. The tab bar at the top includes 'When to Present', 'List Access', 'Tags', 'Rules', and 'Options' (highlighted in blue). The main content area has three sections:

- When Trial is Restarted Due to Feedback:**
  - ☐ Limit the number of restarts to
  - ☐ If using trial variables, get a new level every time  
The same trial levels will be used unless this option is checked
- Trial Pacing:**

Useful for sequencing trials in fMRI and EEG Experiments.

  - ☐ Present next trial  milliseconds after this trial has started  
This does not provide a hard stop. For more information on sequencing and synchronizing trial presentation with fMRI, see this [support note](#).
- ☐ Randomize events in this trial  Groups...
  - ☐ Except the first one
  - ☐ Except the last one

Randomizing events within a trial is a feature that is useful only to a small number of experiments. If your intention is to randomize trials, you can do so in the List Access tab or at the block level.

If you specify that a trial should be restarted in the Event Editor's **Feedback** tab, the options in this tab allow you to limit how many times the trial is repeated, and whether to randomize again.

The Trial Pacing<sup>1</sup> feature comes in handy in fMRI and EEG experiments, especially when an external trigger is not readily available.

---

<sup>1</sup> This feature was renamed to Trial Pacing starting with SuperLab 5.0.3.

# More In-Depth Features

This chapter describes how to:

- Set correct responses
- Provide feedback to participants
- Use tags, and why
- Send event codes to external data recorders, e.g. EEG or an eye tracker

## DETERMINING THE CORRECT RESPONSE

When running an experiment, SuperLab includes in the data file an “Error Code” column that indicates if the participant’s response is correct or not<sup>2</sup>. This column makes subsequent data analysis easier – but only if you tell SuperLab what is the correct response for each trial or stimulus list item.

Specifying a correct response also makes it possible to provide feedback based on correctness of the response, for example to display a message or to increment a counter.

In SuperLab, you can specify a correct response:

- Based on a stimulus list item if the event uses a list
- For the event itself if it does not use a stimulus list
- Based on a tag that is attached to the trial or a stimulus list item

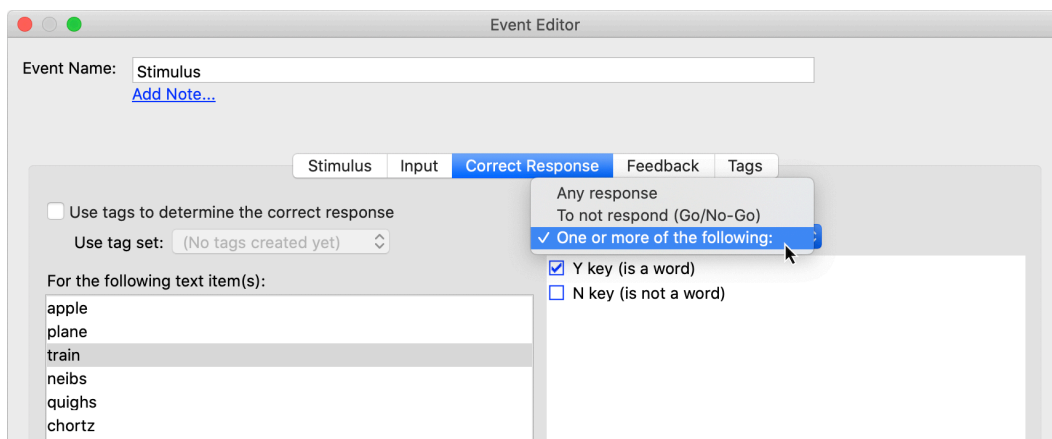
A correct response can be any response that you have specified in the Participant Input dialog (see **Experiment** menu) even if they are from different devices, e.g. keyboard and an RB-730 response pad.

A correct response might be that the subject must not respond. This is an important feature for **Go/No-Go** experiments.

The following screen snapshot shows a correct response being defined for a list item:

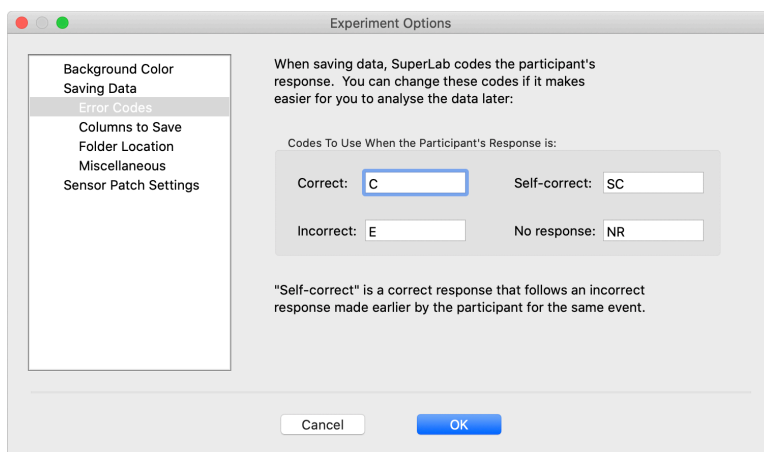
---

<sup>2</sup> The “Error Code” column will also tell you if the participant did not respond, and even whether the response is a self-correct, i.e. they answered incorrectly at first, and then correctly.



## Saving the Responses

The error codes saved by SuperLab can be customized by clicking on **Experiment** menu, then **Options**, then “Error Codes”:

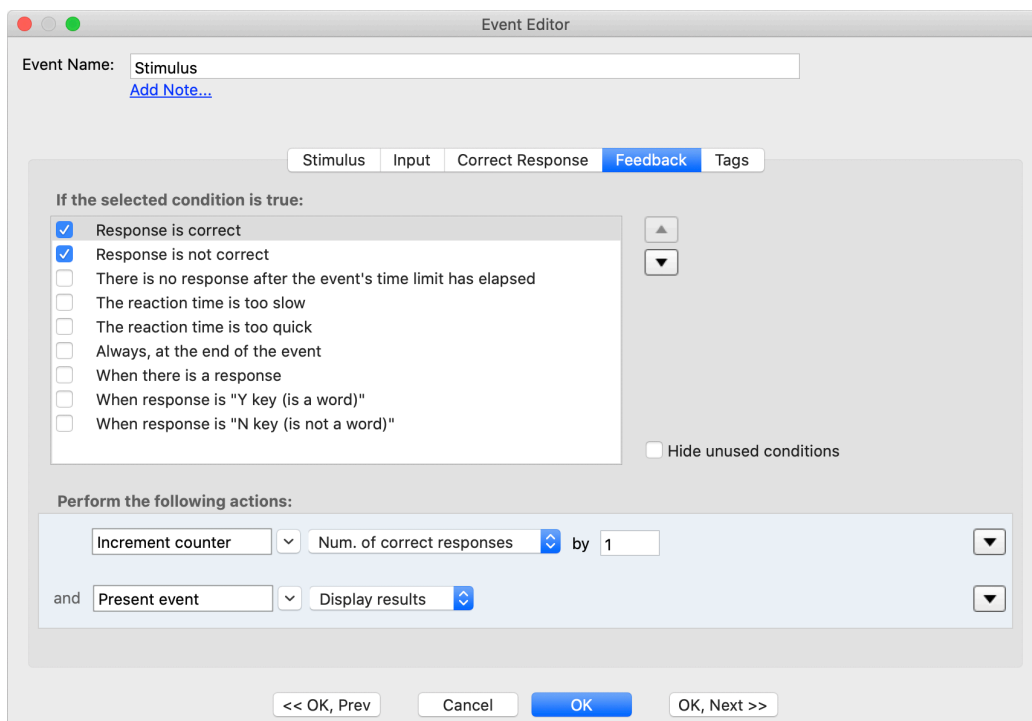


See chapter 7, [Running Experiments & Viewing Data](#), prior to collecting data.

## FEEDBACK

It is common in an experiment to provide feedback to participants, or to want to perform certain actions based on a participant’s response such as incrementing a counter or storing the response for later use.

You can do all these things and more in the Event Editor’s **Feedback** tab, where a feedback consists of a condition and one or more actions that take place if the condition is true:



In the screen snapshot above, note the bottom two conditions:

When response is “Y key (is a word)”

When response is “N key (is not a word)”

In addition to the core conditions (if correct or not, and so forth), SuperLab will list all the responses that you have defined in the Participant Input dialog, allowing you to provide feedback when a specific key is pressed.

As for feedback actions, they work identically to how actions are used in rules. See chapter 5, [Conditional Branching](#).

## TAGS

### Describing a Trial or Stimulus List Item

The simplest use of the tags<sup>3</sup> feature is to “describe” a trial or stimulus list item and help you with the data analysis. But they can also be used to tag trials “on the fly” while an experiment is running.

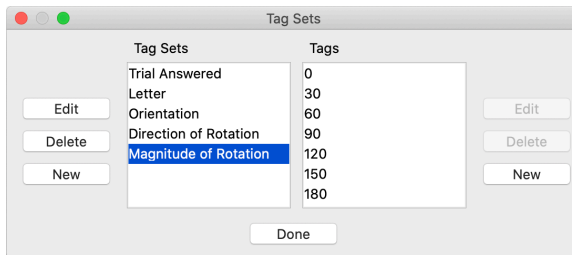
Let’s take the example of a letter rotation experiment. When the data collection is done and it is time to analyze the data, we are interested in the following additional information: which

---

<sup>3</sup> Formerly called codes and code values

letter was presented, orientation (forward or backward), direction of rotation (clockwise or counterclockwise), and angle that it was presented at.

To accomplish this, we create four tag sets and add relevant tags to them (click on **Experiment** menu then **Tag Sets...**):



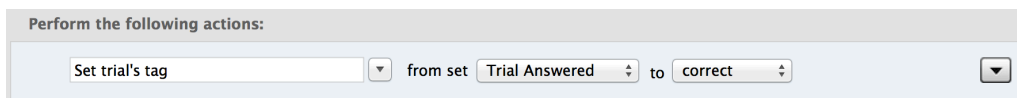
Once the tag sets are created, the next step is to actually tag the trials or stimulus list items. This is done in the **Tags** tab of either the Trial Editor or Stimulus List Editor.

The four tag sets that were created will result in four additional columns (fields) being added to the data file. When the participant responds, a row gets added to the data file and SuperLab will save the tags that you have applied to that trial or stimulus list item.

Taking time to tag your trials can be helpful when analyzing your data.

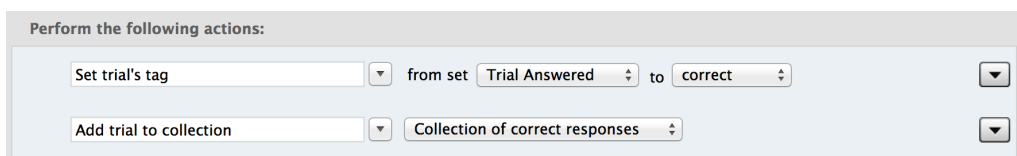
## Tagging On The Fly

It is possible to change how a trial is tagged on the fly while an experiment is running. This is done using a feedback action or a rule action:



An example is when an experiment requires that all trials in a block be answered correctly before proceeding to the next block. To accomplish this, you use an action to tag a trial based on the participant's response. See the section [Living Without Rules](#) in chapter 5 for an example that describes how to do this.

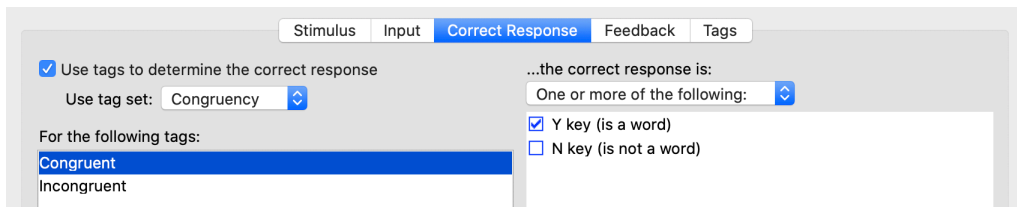
Another example is when you want to count the number of trials that have been tagged a certain way over the course of the experiment. You can accomplish that by using a parameter of type Collection of Trials (click on **Experiment** menu then **Parameters...**) and then add the trial to it:



If your experiment does require anything like this, it is highly recommended that you read chapters 4 and 5, [Parameters](#) and [Conditional Branching](#), respectively.

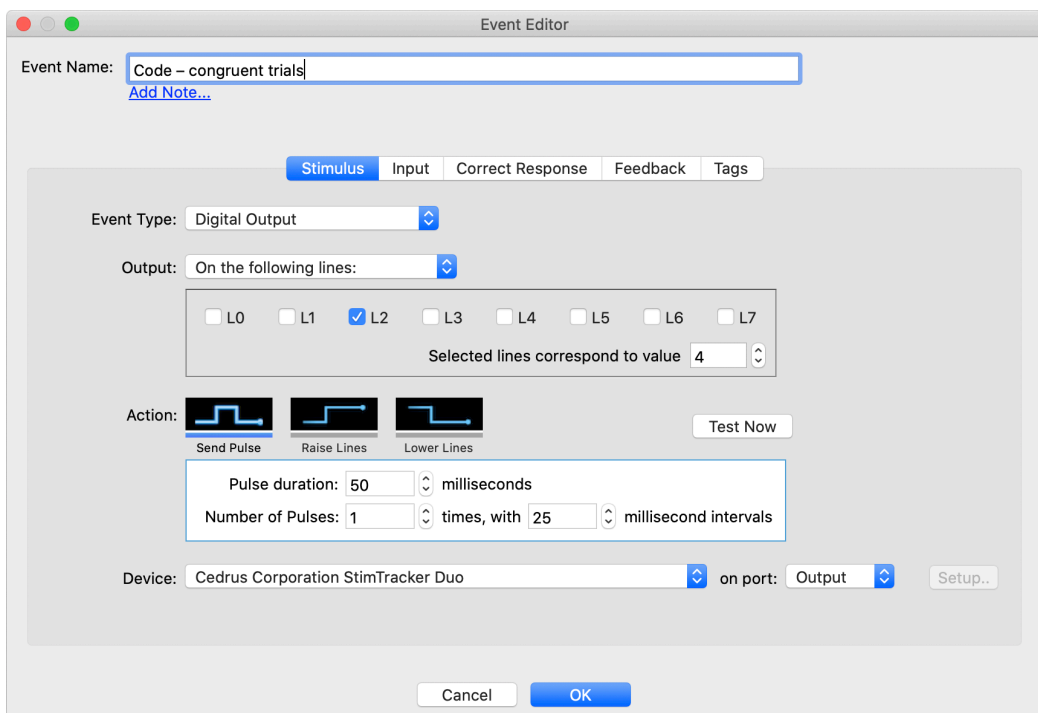
## Using Tags to Determine Correct Responses

There are cases where an experiment works best if the participant's correct response is determined based on how a list item is tagged. This is done in the Event Editor's **Correct Response** tab:



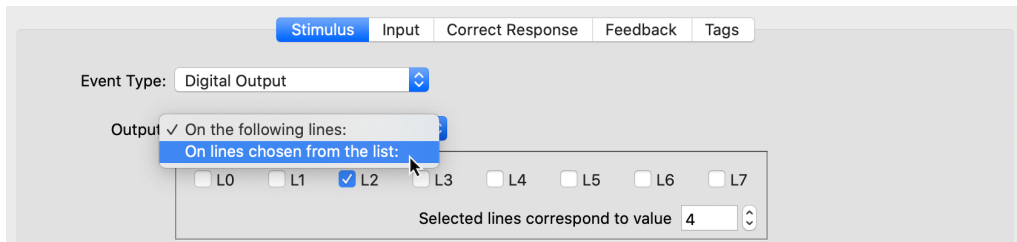
## SENDING EVENT CODES

Many experiments require that an event code be sent to EEG/ERP, eye trackers, or other recording equipment. In SuperLab, this is done by creating an event and setting the Event Type to “Digital Output” in the Event Editor’s **Stimulus** tab:



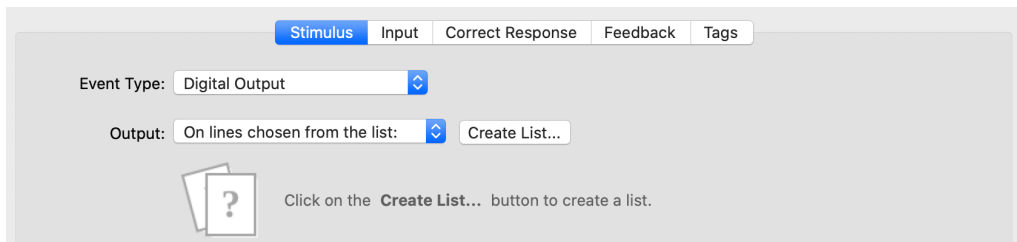
You select the device at the bottom, e.g. StimTracker or an I/O card. You can specify the event code by entering it directly or by enabling the corresponding checkbox.

You can use lists with “Digital Output” event types as well:



The screenshot shows the 'Event Editor' interface with the 'Stimulus' tab selected. The 'Event Type' is set to 'Digital Output'. The 'Output' dropdown is open, showing 'On the following lines:' and 'On lines chosen from the list:'. Below this, a row of checkboxes for lines L0 through L7 is shown, with L2 selected. A text field below the checkboxes indicates 'Selected lines correspond to value 4'.

The list editor can be invoked directly from within the Event Editor by clicking on the **Create List** button, or you can create lists by clicking on **Experiment** menu then **Stimulus Lists...**



The screenshot shows the 'Event Editor' interface with the 'Stimulus' tab selected. The 'Event Type' is set to 'Digital Output'. The 'Output' dropdown is set to 'On lines chosen from the list:'. A 'Create List...' button is visible next to the dropdown. Below the button, there is a question mark icon and the text 'Click on the **Create List...** button to create a list.'



# Parameters

Say you have been asked to organize and referee a sports game. During the play, you need to keep a running total of the number of points scored by each team, note the names of game participants<sup>4</sup> that score points, keep track of remaining time or turns, and number of fouls, injuries, and replacements.

These are called **parameters**. When the game starts, you adjust the parameters depending on certain **conditions** and **criteria**. For example:

- You increase the “points scored” parameter when a team puts the ball in the goal.
- You increase the “number of fouls committed” parameter when a game participant commits a foul.
- If a **rule** dictates so, you eject the participant from the game after he or she has committed a certain number of fouls.
- You call an end to the game when the time is up or after a certain number of turns.

You will have too many game parameters, so you choose to display only a few of them on the scoreboard for all to see. Others are not displayed but used nevertheless, e.g. to give an award to the game participant that scores the most points.

In SuperLab, we use the word *experiment* instead of *game* but you can set up parameters as well, e.g. to keep counts, store responses and text, or keep track of presented trials. You can display some of them to the participant or use them with rules to determine any number of things, e.g. what block to present next.

Not all experiments require parameters, but when they do, they are easy to use in SuperLab. There are three steps:

① **Create it** → ② **Modify it** → ③ **Use it**




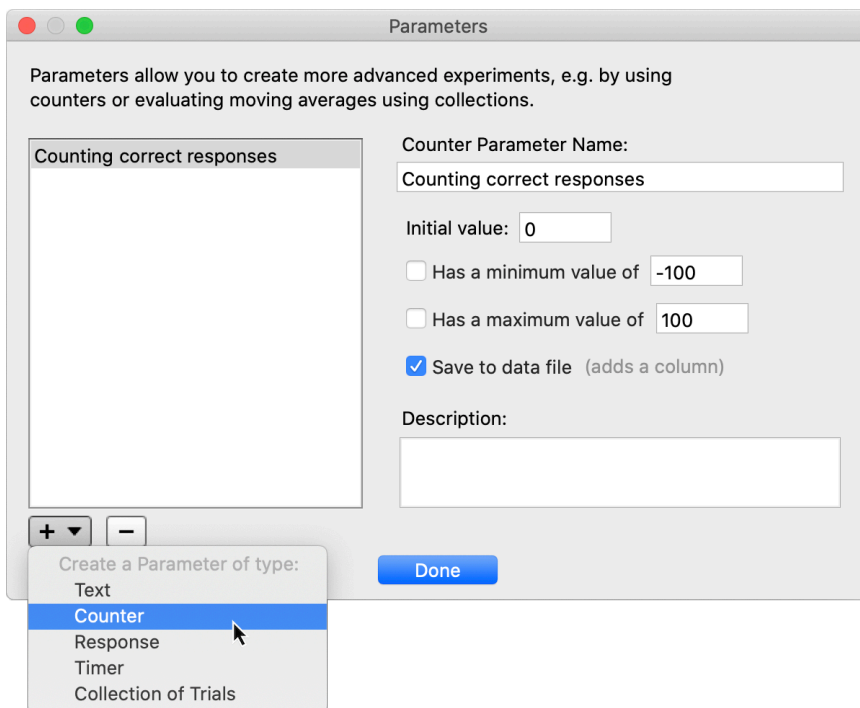
<sup>4</sup> Also known as “players”!

## A TASTE OF PARAMETERS

Let's start with an example that displays to participants the number of correct responses that they have made.

### ① Create the parameter

- Click on the **Experiment** menu and choose **Parameters**
- Click the **Add** icon  icon and choose **Counter** from the menu
- Change the name to “Counting correct responses”
- Click **Done**



### ② Modify it

In this step we want to count correct responses by adding 1 each time a participant makes a correct response.

- In the main experiment window, double-click on the event that collects the participant's response and has a correct response defined
- In the Event Editor, click on the **Feedback** tab
- Enable the checkbox for condition **The response is correct**

- Select the action **Increment Counter**; SuperLab defaults to the parameter you created and a value of 1

Perform the following actions:

Increment counter    Counting correct responses    by 1

- Click **OK** to close the Event Editor

### ③ Use it

- In the main experiment window, double-click on the event that you want to display the result
- Click the **Stimulus** tab
- Type in the field “**You got**”
- Click on **Insert Parameter** button and choose parameter “Counting correct responses” from the menu
- Type “**right.**”

Event Editor

Event Name: Display results – the end

[Note:](#) This event displays the number of correct responses made by a participant.

Stimulus   Input   Correct Response   Feedback   Tags

Event Type: Text

Use Text: Text That I Enter Below

Insert Parameter

You got \$(Counting correct responses) right.  
Thank you for participating in this experiment.

Font...  
Settings...  
Preview  
☒ Show grid

<< OK, Prev   Cancel   OK   OK, Next >>

- Click **OK** to close the Event Editor

- Click the **Save Experiment** icon  and run the experiment

When the event is presented, SuperLab will display how many correct responses were made by the participant.



**Important:** In the example above, the event that \*displays the parameter value should be in a separate trial than the event that modifies the parameter

## RECAP OF THE THREE STEPS

In our example above, you created your first parameter, modified it, and used it. In more general terms:

- ① Parameters are always *created* by choosing **Parameters** from the **Experiment** menu. The only exception is built-in parameters, described later in this chapter.
- ② Parameters can be *modified* by an action in the Trial or Block Editor's **Rules** tab, or the Event Editor's **Feedback** tab.
- ③ Parameters can be *used* as criteria in the Block Editor or Trial Editor, in either their **Rules** or **When to Present** tabs. They can also be used in the Event Editor's **Stimulus** tab as we saw in the example above.

## PARAMETER TYPES

There are five types of parameters in SuperLab:

<b>Text</b> Stores a text string of any size	Useful for capturing participant typed responses, e.g. for thought recall, capturing a sequence, and anything involving text.
<b>Counter</b> A numeric value, optionally with a minimum and/or a maximum value	Useful for counting anything, e.g. number of correct responses or money (for experiments involving rewards). Also useful for randomization, e.g. to simulate rolling of dice.
<b>Response</b> Stores a participant's response	To recall a participant's response and be able to compare it to another predefined response
<b>Timer</b>	Useful for keeping track of elapsed time and performing an action when a certain time is exceeded
<b>Collection of Trials</b> Can keep track of any number of trials; see below for more details	Keeps track of presented trials; see below

The last parameter type, “Collection of Trials”, is very powerful when needed, for three reasons.

First, *you*, the experimenter, get to decide exactly which trials get added to the collection. These may be only the trials that were answered correctly, or trials that are tagged a certain way.

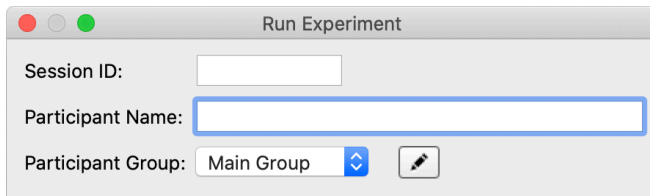
Second, when the time comes to make a decision based on what’s in the collection, there are a number of useful options available:

And third, you can easily implement a moving window paradigm by limiting the size of your collection. See section [A Memory Experiment](#) in the next chapter for an example of how a Collection of Trials parameter can be used.

## BUILT-IN PARAMETERS

In addition to creating your own parameters, SuperLab contains some built-in ones; you can see them in the Event Editor’s **Stimulus** tab when the stimulus type is set to Text. If you have created your own parameter, they appear first, followed by the built-in ones:

The “Participant Name” parameter is what you type (or the participant types) just prior to running an experiment:



A screenshot of a 'Run Experiment' dialog box. It features three standard macOS window control buttons (red, yellow, green) in the top-left corner. The title bar reads 'Run Experiment'. Below the title bar, there are three input fields: 'Session ID:' with a small rectangular text box, 'Participant Name:' with a larger rectangular text box that has a blue border, and 'Participant Group:' with a dropdown menu showing 'Main Group' and a blue arrow icon. To the right of the dropdown menu is a small icon of a pencil inside a square box.

## The Power

And last but not least, the power of parameters is multiplied when combined with rules; see the next chapter.

# Conditional Branching

Instead of presenting trials straight through from beginning to end, many experiments require the ability to control the flow of the experiment. Examples:

- Prevent experiment from proceeding to next block until all trials in the current block have been answered correctly.
- Present trials in a block until the moving average of the participant's reaction time drops under a certain threshold, e.g. in memory experiments.
- Alter the order in which trials are presented based on the participant's response.

In SuperLab, conditional branching refers to a number of features that allow you to do all these things and a lot more. One of these features is **rules**, but you can accomplish a fair number of things even without rules.

The rest of this chapter proceeds with two design examples.

## LIVING WITHOUT RULES

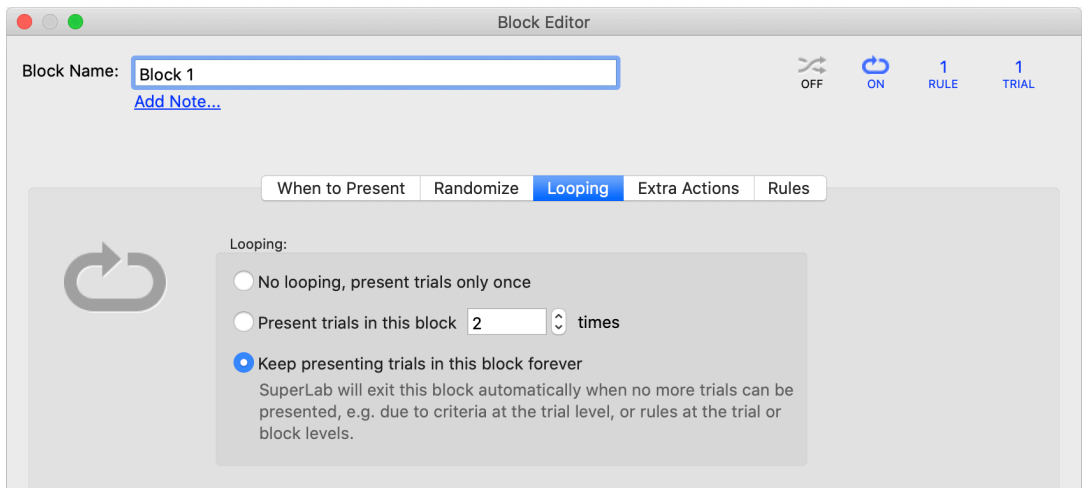
Let's look at the first experiment listed as an example above: we want to make sure that all trials in the current block have been answered correctly before moving on to the next block. The basic approach for designing the experiment is this:

1. When all the trials in the block have been presented, some may not have been answered correctly. We need to repeat the block to give participants another chance.
2. When the participant responds correctly, we need to tag the trial as answered correctly.
3. We need a criterion that tests if the trial has already been answered correctly to avoid presenting it again.

This experiment can be implemented without using the rules feature.

### Step 1: Repeating the Block

In the Block Editor, click on the **Looping** tab and change the option to **Keep presenting trials in this block forever**.



This option makes it sound like your participants will be sitting at the computer and answering trials past midnight, but that is not the case: remember that we are going to add a criterion that prevents a trial from being presented again if answered correctly. When SuperLab realizes that no more trials are being presented, it automatically exits the block to go to the next one.

## Step 2: Tagging Trials

You can tag trials by going to the Trial Editor. If you need to tag particular trials while the experiment is running, you want to set this up using an action. In our example, we will do so in the Event Editor's **Feedback** tab.

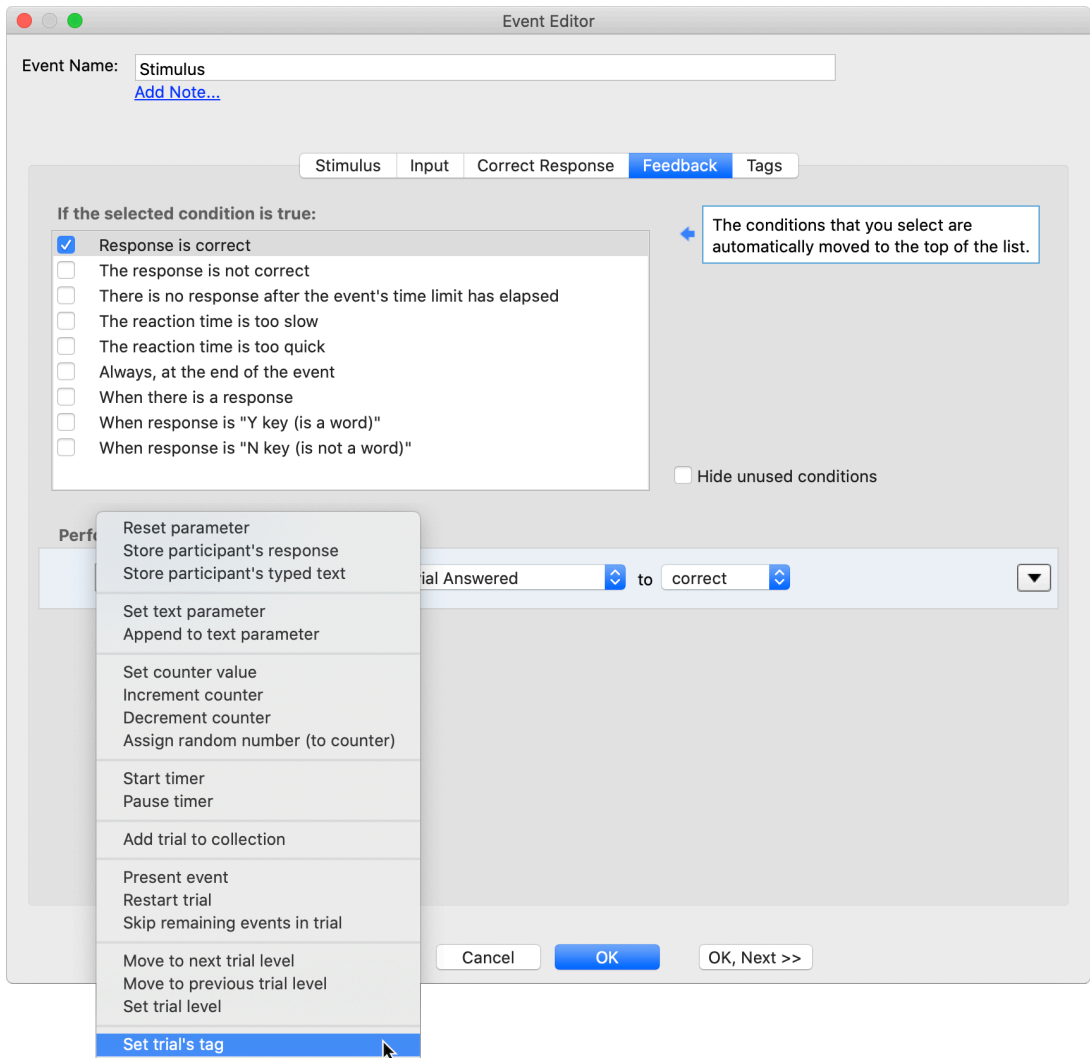
Provided that you have defined a correct response in the **Correct Response** tab:

- Click on the Event Editor's **Input** tab.
- Click on the checkbox **After any response from the participant**.
- Click on the Event Editor's **Feedback** tab.
- In the list of conditions, turn on the checkbox for the condition **The response is correct**. The bottom part of the dialog becomes enabled, with a default action.
- Click on the list of actions and choose **Set Trial's Tag**.
- If not selected already, select tag set "Trial Answered" and tag "correct".

You can create more actions by clicking on the down arrow icon .

The tag set "Trial Answered" is created for you automatically with every new experiment because it is common to want to tag a trial as answered correctly while running an experiment.

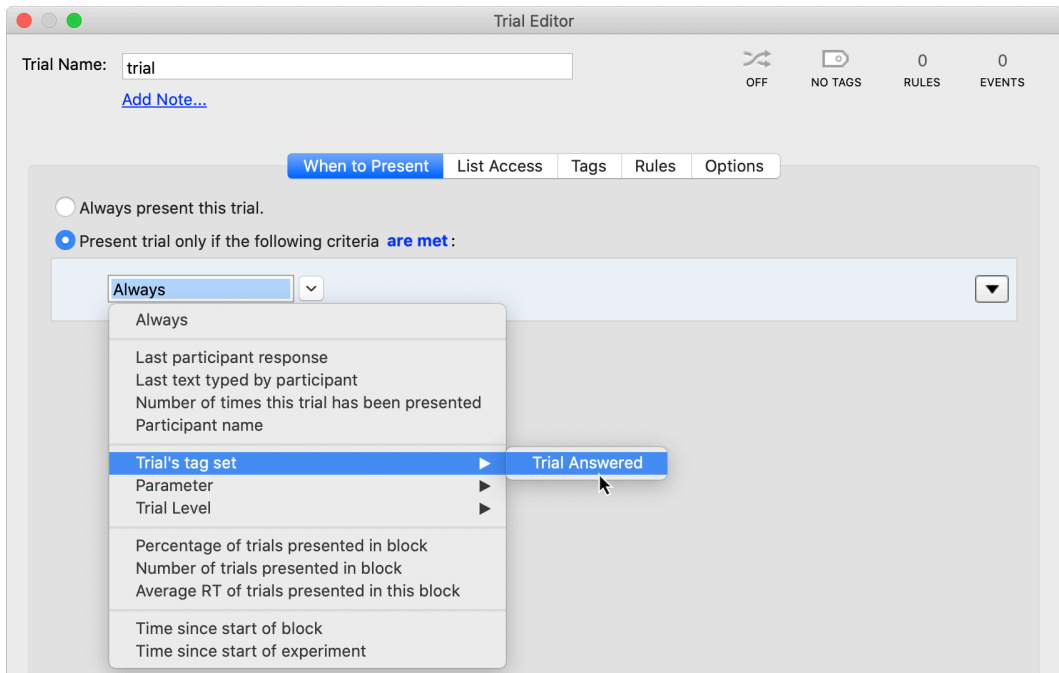




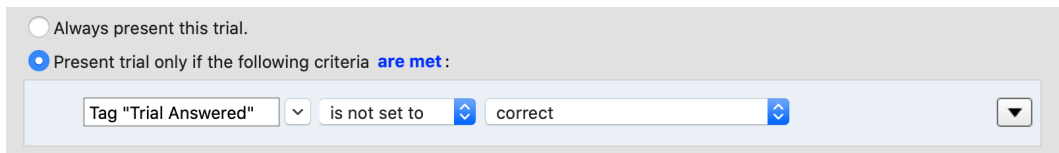
### Step 3: Criterion

The last step is to prevent a trial from being presented if it has already been answered correctly. We do so in the Trial Editor's **When to Present** tab:

- Click on the radio button **Present trial only if the following criteria are met**.
- Click on the criteria menu.
- Click on the "Trial's tag set" menu item; a sub-menu appears showing you all the tag sets that you have created.
- Select the tag set "Trial Answered".



When done, the criterion should look like this:

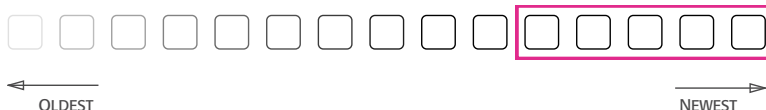


That's it. These three simple steps are all it takes to make sure that we get a correct response for all trials before moving on to the next block.

## A Memory Experiment

In this experiment, we want to keep presenting trials in a block until the moving average of the participant's reaction time drops under a certain threshold, e.g. 1500 milliseconds. It is a moving window paradigm where we consider only the last N trials; in this example, N will be 20.

For this design, a key feature is **parameters**, discussed in the previous chapter, but here we elaborate on one particular type called "Collection of Trials". A collection parameter does what it implies: it keeps track of whatever trials you add to it. You can add as many trials as you want, but here is the nifty thing: you have the option to limit the size of the collection to, say, 5 trials. Once the collection has 5 trials, then every new trial that you add will cause the oldest one to be removed from the collection:



In essence, you can easily obtain a moving average by simply enabling the option to limit the size of the collection.

The basic approach for designing the experiment is:

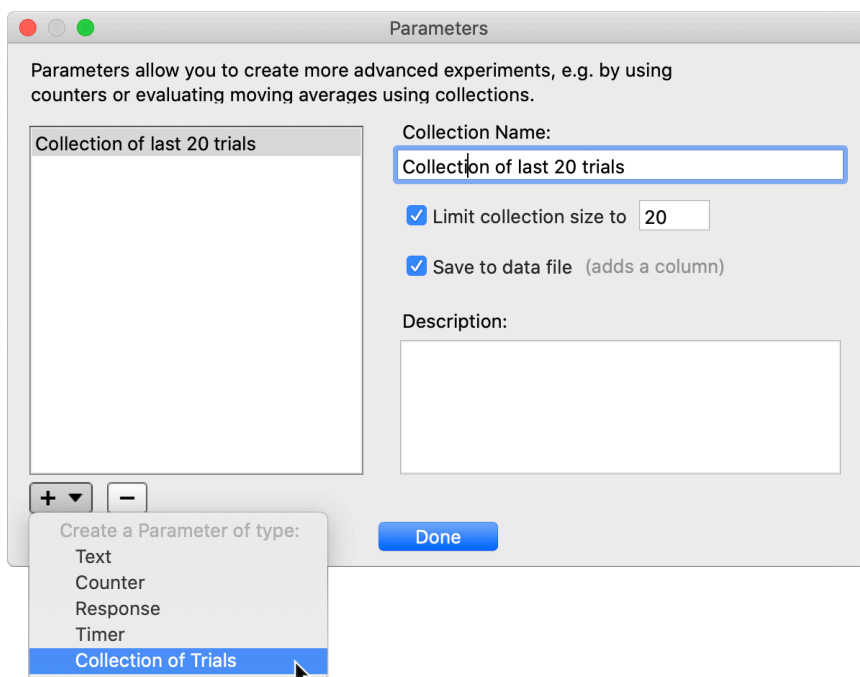
1. When all the trials in the block have been presented, the average reaction time may not be below our desired threshold yet, so we need to repeat the block.
2. Create a **parameter** of type “Collection of Trials”.
3. Create a feedback action that adds a trial to the collection.
4. We need a rule that tests if the average reaction time of the trials in the collection has dropped below the desired threshold, and if true, exits the block.


### Step 1: Repeating the Block

In the Block Editor, click on the **Looping** tab and change the option to **Keep presenting trials in this block forever**, just as we did in the earlier example.

### Step 2: Creating the Collection Parameter

Click on the **Experiment** menu and choose **Parameters**. A dialog appears:

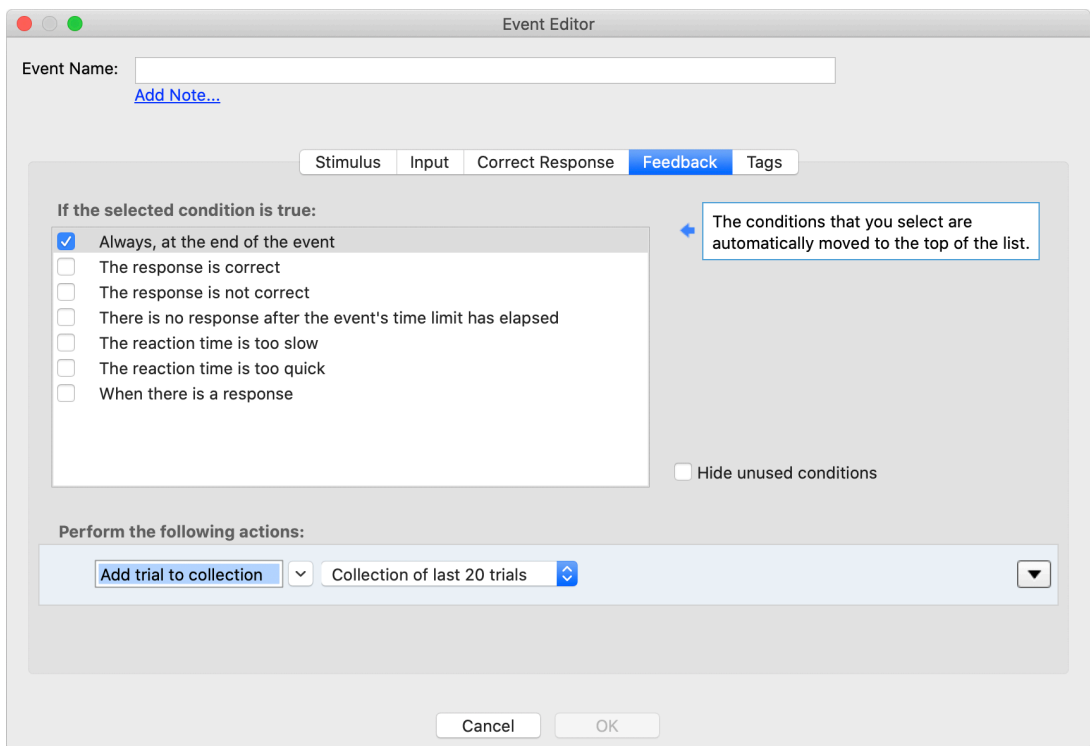


In that dialog, click on the **Add** icon  and choose “Collection of Trials” from the menu.

That’s it; you now have a new collections parameter. Give it a meaningful name, e.g. “Collection of last 20 trials”, and make sure that the **Limit collection size** checkbox is turned on.

### Step 3: Create Feedback Action

The next step is to add presented trials to the collection. In the Event Editor, click on the **Feedback** tab and select the condition **Always, at the end of the event**. At the bottom of the dialog, change the action to **Add trial to collection**. On the right side, you can select another collection if you have more than one.

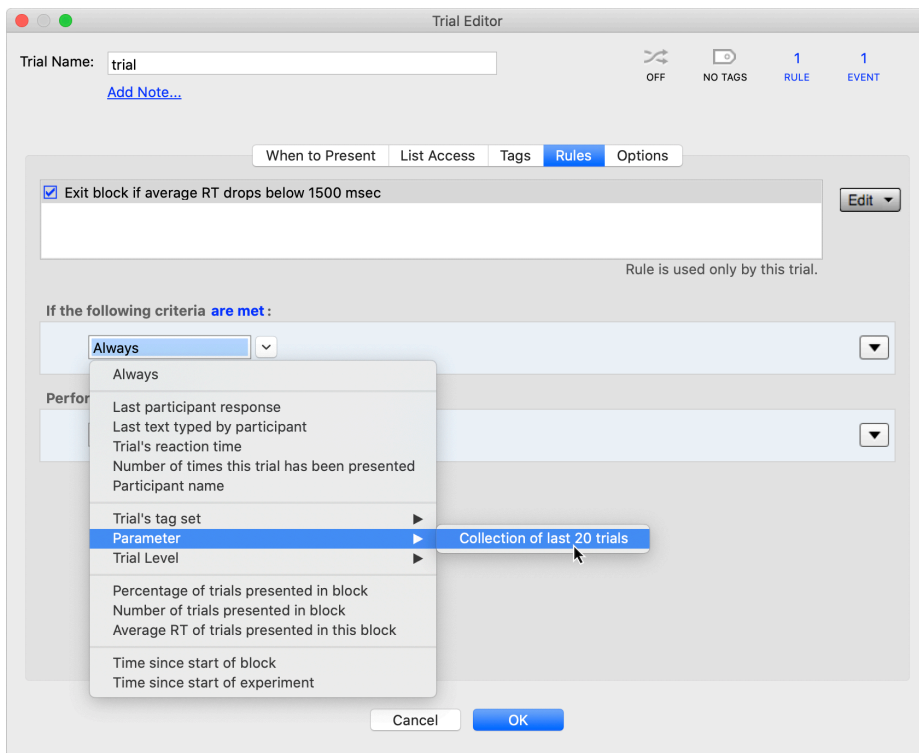


### Step 4: Looking for the Exit

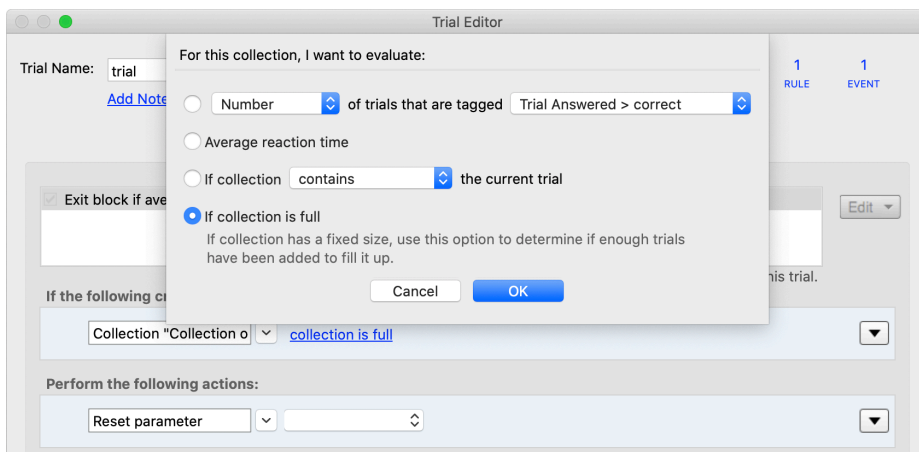
We have created the collection and added presented trials to it. Let’s create a rule to determine when the average reaction time has dropped below 1500 milliseconds. In the Trial Editor’s **Rules** tab:


- Click on the Edit menu  and select **Create Rule**.

- SuperLab will prompt you for the rule's name; type “Exit block if average RT drops below 1500 msec”. Click on **OK**. It is always a good idea to give a meaningful name to the experiment's parameters, rules, trials, and so forth.
- Click on the criterion menu and select “Parameter”. A list of parameters appears in a submenu. We only have one in this case, “Collection of last 20 trials”; select it.



- There are several collection attributes that you can test for; SuperLab will automatically prompt you to choose one. Select **If collection is full** and click on **OK**.

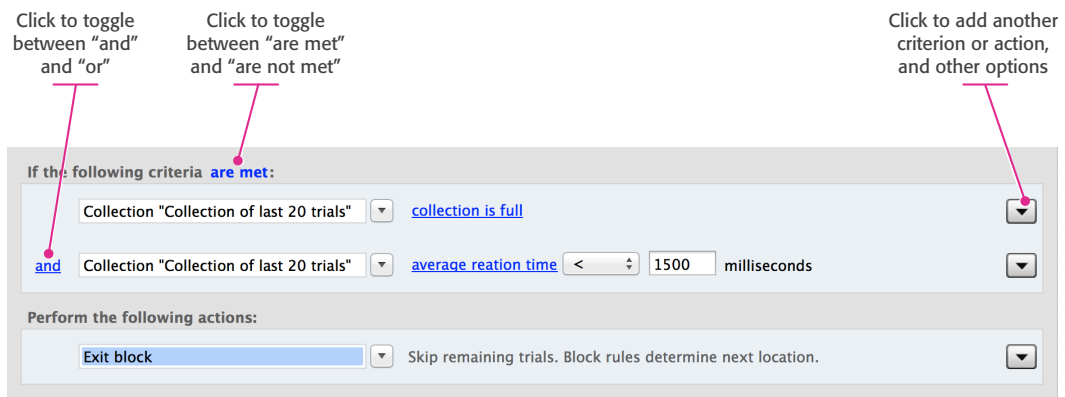


- Click on the down arrow icon  and select **Add Criterion**.
- Click on the newly added criterion and select “Parameter” and then “Collection of last 20 trials”, as we did earlier.
- SuperLab will again prompt you to choose what you want to test for; this time, select **Average reaction time** then click on **OK**.
- In the edit field, type “1500”.



- Change the action to “Exit block”.
- Click on the **Apply** button.

Here is what your rule should look like, along with some notes:



Note that we created *two* criteria, not one. The first one makes sure that the collection is full, meaning that we need to have at least 20 trials presented in addition to having an average RT that is less than 1500 milliseconds. But your own experiment’s requirements may be different; feel free to adapt the example above to your needs.

## The Rules About Rules

You have just created your first rule. A rule consists simply of one or more **criteria**, and if the criteria are met (or not met), one or more **actions** will be performed.

The rule editor is simple to use but is also very powerful. Here are some important facts:

- **Most important:** a trial rule is executed *after* a trial has been presented; a block rule is executed *after* a block has been presented.
- Rules will always be applied *before* looping and any options that you may have selected in the Block Editor’s **Looping** and **Extra Actions** tabs.

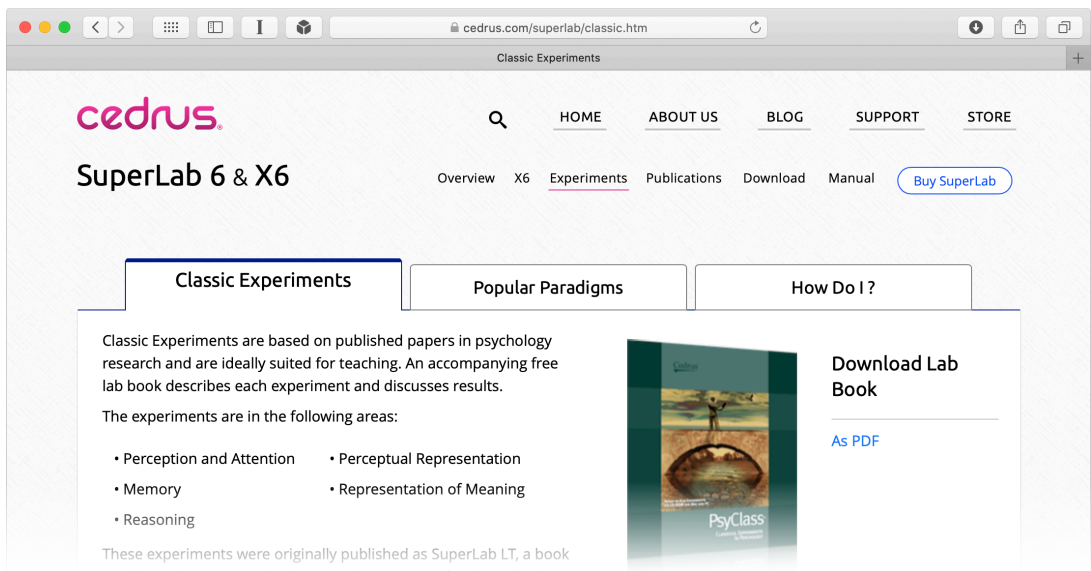
- Rules are created and edited in the Block Editor or Trial Editor's **Rules** tab.
- A trial or block can have many rules. They are executed in the order in which they appear in the editor. This can be important, e.g. if a rule contains an action to go to another block, then subsequent rules will be skipped. You can change that order.
- Rules created in the Trial Editor can only be used with trials, not blocks. And vice versa.
- A trial rule can be used with any number of trials; same for blocks.

## OTHER APPLICATIONS FOR RULES

The power of rules can be vastly extended when combined with **parameters**, **feedback**, **tags**, and **trial levels**.

## LEARN MORE

The best place to learn more is our website. There are three experiment libraries: Popular Paradigms, Classic Experiments, and How Do I:







# Randomization

There are three main areas of randomization in SuperLab:

- You can randomize the order in which trials are presented.
- You can create randomized subsets of a stimulus list.
- You can vary an aspect of a presented trial, for example the location of a stimulus on the screen or the text color.

These three randomization areas are independent of each other: you can use either one or all at the same time.

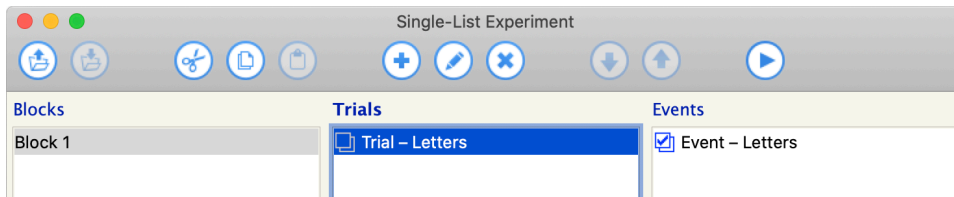



Reminder: you should use stimulus lists in your experiment whenever possible.

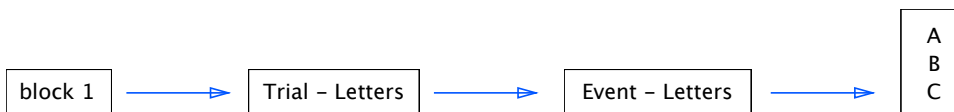
## Randomizing The Order in Which Trials Are Presented

### AN EXAMPLE USING A SINGLE STIMULUS LIST

The following basic experiment uses a single stimulus list:

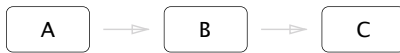


The “Event – Letters” event uses a stimulus list, as noted by the stacked squares icon . But we cannot tell how many items the list uses; going forward we will use the following style diagrams instead:

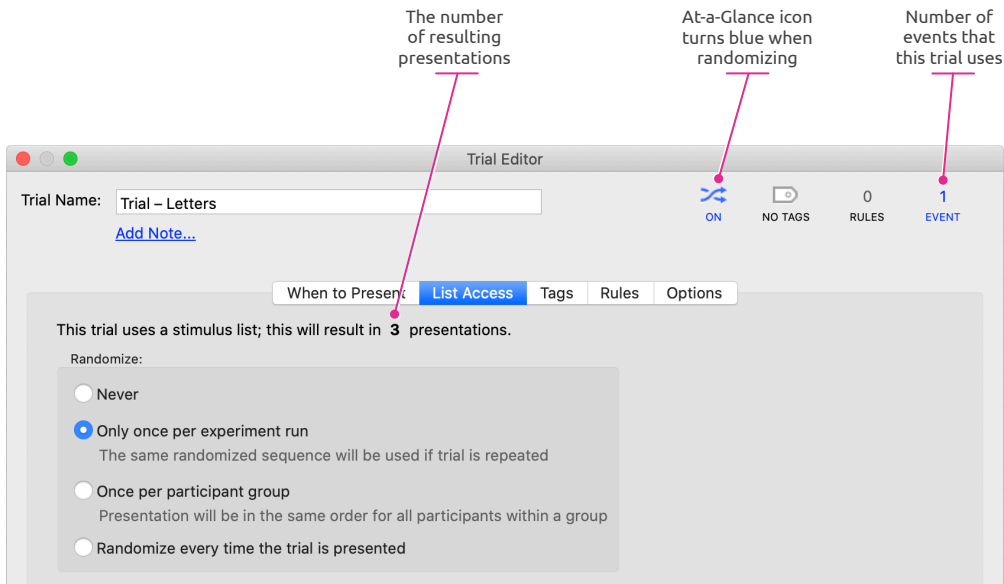



The diagram shows that “block 1” is linked to, i.e. uses “Trial – Letters”, and “Trial – Letters” uses “Event – Letters”, which in turn uses a list that has three items: A, B, and C. This will result in three presentations of the trial, each presentation using one of the list items.

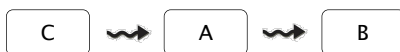
By default, none of the randomization options are turned on when you create a new experiment. When you run it, you will get the following presentation sequence:



To randomize, double-click on the trial to edit it. In the Trial Editor, click on the **List Access** tab and select the option **Only once per experiment run**:

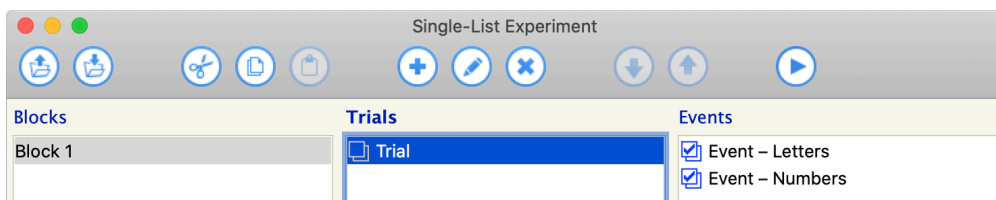


When you click on **OK** and run the experiment again, the order of trial presentation will now be random. A squiggly arrow  indicates randomness:

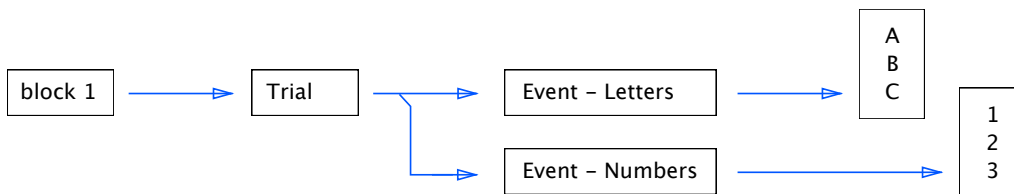


## USING TWO STIMULUS LISTS IN THE SAME TRIAL

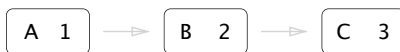
Now let's move into a more fun experiment involving two lists in the *same* trial:



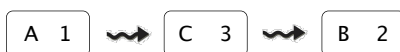
Using our diagram notation, the experiment can be represented as follows:



In other words, “block 1” is linked to “Trial”, which in turn is linked to two events that each uses a stimulus list. Assuming the experiment is designed in such a way that a letter and a number appear at the same time on the screen<sup>5</sup>, and that we started with a brand new experiment where no randomization options have been enabled yet, we obtain the following presentation sequence:



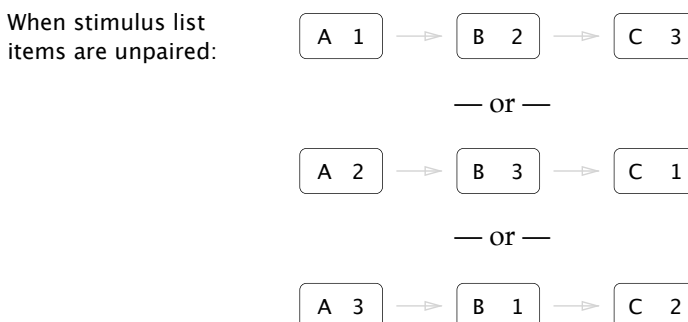
To randomize, double-click on the trial to edit it. In the Trial Editor, click on the **List Access** tab and select the option **Only once per experiment run**, as shown in the screen snapshot on the previous page. When you run the experiment again, the order of trial presentation will now be random:



Note that the stimulus list items are **paired**. Thus far, we are only randomizing the pairs, but “A” will always appear with “1”, “B” with “2”, and “C” with “3”.

SuperLab offers two additional methods of arranging the stimulus list items: unpaired, and crossed. These options are also found in the Trial Editor’s **List Access** tab.

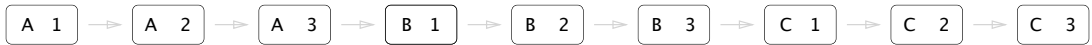
When **unpaired**, a letter from the first stimulus list is combined at random with a number from the second list. When you run the experiment, you will get *one* of the following presentation sequences:



<sup>5</sup> To make two visual events appear at once, set the first event to “End immediately” (**Input** tab) and the the second event to *not* erase the screen (**Stimulus** tab, then click on **Settings** button).

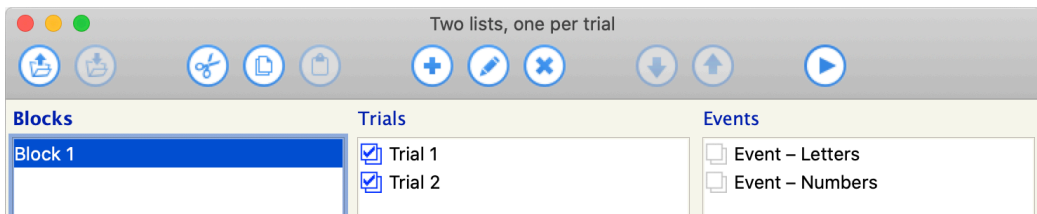
When **crossed**, a letter from the first stimulus list is combined with *every number* from the second list. The total number of presentations is no longer 3; it is the number of items in the first list *multiplied* by the number of items in the second list, or 9 presentations in our example:

When stimulus list items are crossed:

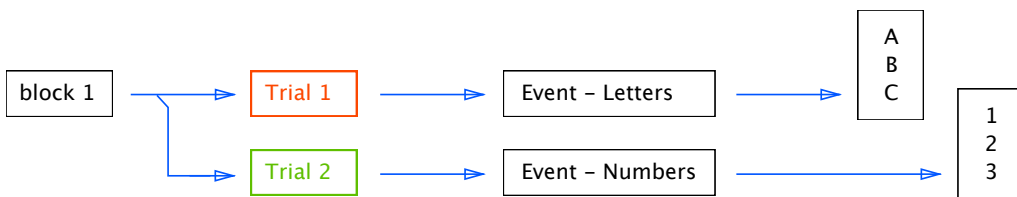


## USING STIMULUS LISTS IN DIFFERENT TRIALS

In this section, we will also use two stimulus lists, but in *different* trials:



Or, using our diagram notation:



For discussion's sake, we represent the trials in different colors. When starting with a brand new experiment where no randomization options have been enabled yet, we obtain the following presentation sequence:



We can randomize the presentation in different ways: at the trial level, at the block level, or both. Let's start with something already familiar: enabling randomization at the trial level:

- Double-click on "Trial 1"; the Trial Editor appears.
- Click on the **List Access** tab.
- Select the option **Only once per experiment run**.
- Click on **OK** to close the Trial Editor.
- Repeat the steps above for "Trial 2".

When running the experiment, we will get the following presentation:



Note the order of letters in “Trial 1” has been randomized, as was the order of numbers in “Trial 2”. But “Trial 1” presentations will always precede those of “Trial 2”.

To randomize the order of the trials, double-click on the block, select the **Randomize** tab, and enable the option **Randomize the trials in this block**. This will yield one of two presentations:



Or



In other words, SuperLab will randomize the letters and the numbers, and the order in which they appear. But in this example, the presentation of letters will be grouped together, as will the presentation of numbers.

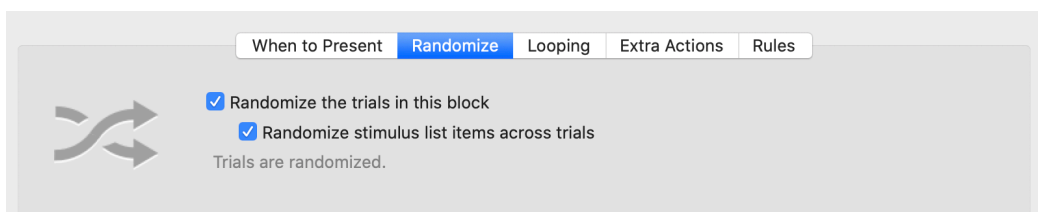
But what if we want to intersperse the letters and numbers, like this:



Trial-level randomization gets priority over block-level randomization. To fully intersperse the trials as shown, we need to turn off randomization at the trial level so that the block can do all the randomization work itself:

- Double-click on “Trial 1”; the Trial Editor appears.
- Click on the **List Access** tab.
- Select the option **Never**.
- Click on **OK** to close the Trial Editor.
- Repeat the steps above for “Trial 2”.

Now double-click on the block. In the Block Editor’s **Randomize** tab and enable the option **Randomize stimulus list items across trials**:



## Sublists

Another method for organizing your experiment is to use sublists. An example is memory experiments where you have a stimulus list of, say, 100 items. In one block, you want to present half of the list items at random, and in the next block you want to show all 100 items and ask the participant if she has previously seen the item.

Creating sublists is straightforward. Once you have created a list and added items to it:

- Click on the **Sublists** tab in the Stimulus List Editor.
- Enable the option **Divide list into**. You will be able to specify the number of sublists and the percentage of items that each sublist should have.

The screenshot shows the 'Text Stimulus List Editor' window with the 'Sublists' tab selected. The 'Stimulus List Name' is 'Letters 1'. The 'Divide list into' checkbox is checked, set to 2 sublists, with a selection method of 'Random, per Experiment'. A table shows 'SubList 1' and 'SubList 2', each with 6 items (50% of the list). To the right, 'SubList 1' is named, contains 6 items, and has 50.00% of the list items. There is an unchecked checkbox for 'Tag items in this sublist as'.

All Stimulus List items are used by sublists.		
SubList 1	6 item(s)	50%
SubList 2	6 item(s)	50%

Name: SubList 1 Contains 6 items  
Has: 50.00 % of list items  
☐ Tag items in this sublist as

Once sublists have been created, you can use them anywhere in SuperLab that you are able to use a regular list.

You can have a tag applied automatically to items in a sublist:

This screenshot is similar to the previous one but shows the 'Tag items in this sublist as' dropdown menu open. The menu has two options: 'Is Training Sublist : Yes' (which is selected with a checkmark) and 'Is Training Sublist : No'.

All Stimulus List items are used by sublists.		
SubList 1	6 item(s)	50%
SubList 2	6 item(s)	50%

Name: SubList 1 Contains 6 items  
Has: 50.00 % of list items  
☒ Tag items in this sublist as

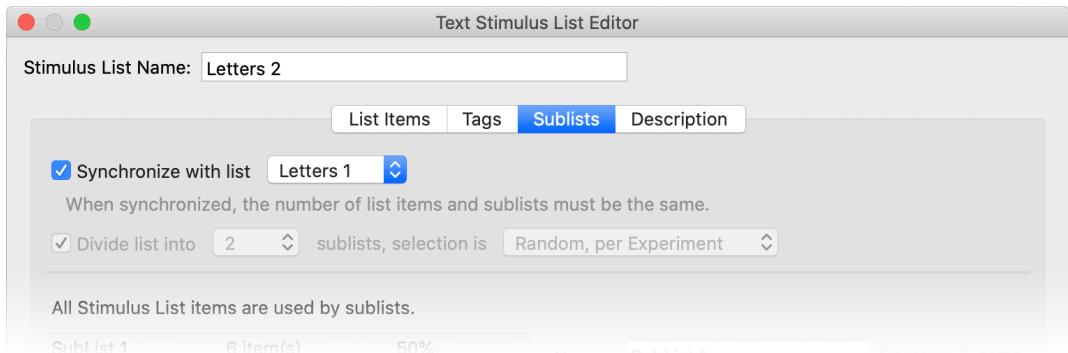
- Is Training Sublist : Yes
- Is Training Sublist : No

This helps in a couple of ways:

- You can use tags in the Event Editor's Correct Response tab, e.g. to have the correct answer be determined by whether the item was presented earlier or not.
- You can use tags in rules.
- Tags are saved in the data file; this makes your data analysis a bit easier later on.

## Synchronizing

A more advanced feature is the ability to synchronize sublists in different lists:



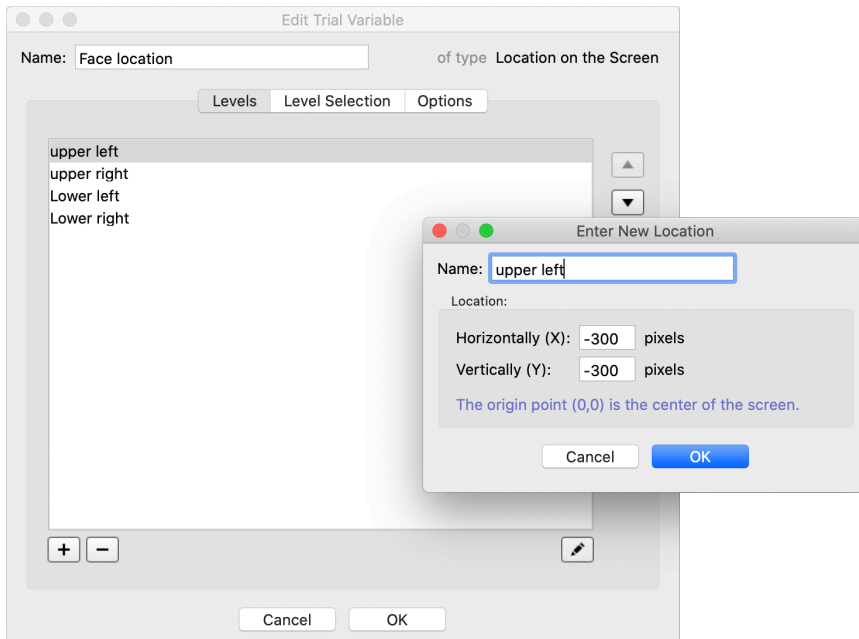
The synchronized lists do not have to be of the same type, e.g. one could be a list of words and the other a list of pictures. But they have to have the same number of items and the same number of sublists.

## Varying an Aspect of a Presented Trial

We've discussed thus far how to randomize the order in which trials are presented. You can also vary the following aspects of a trial itself:

- **Location** of a visual stimulus
- **Text color**
- **Time limit**, which can be used to limit the time provided for a participant to respond and/or the exposure time for a visual stimulus

To use these capabilities, you create a **trial variable** by clicking on the **Experiment** menu and choosing **Trial Levels**. Once created, you can add as many **trial levels** as you need:



The previous screen snapshot shows a variable of type Location that has four levels named “upper left”, “upper right”, and so forth.



Choose meaningful names for your trial levels. These names are saved with each row in the data file (except for variable type Time Limit, which does not take names - the time limit value is saved instead).

SuperLab provides four different methods for selecting levels: sequentially, random with replacement, random with no replacement (default), or manual. The first three are self-explanatory. The fourth one, manual selection, deserves more attention; see “Manual Trial Level Selection” below.

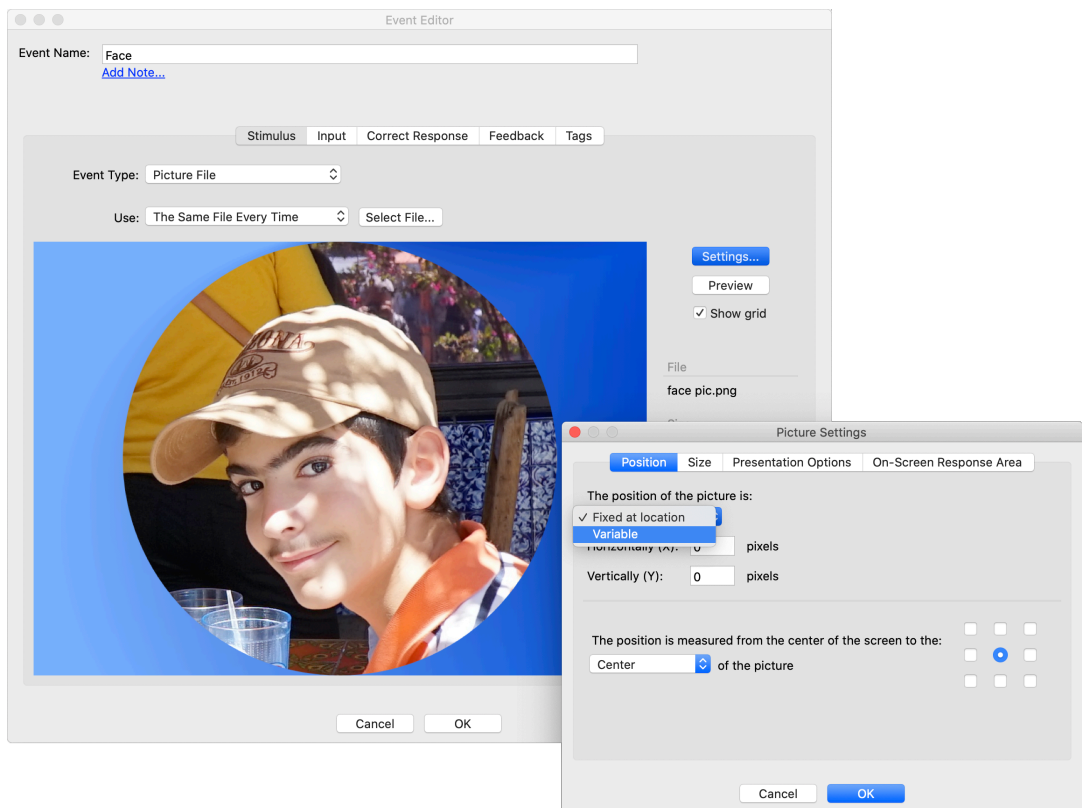
Here is how to use a trial variable after it has been created.

## VARYING BY LOCATION

Trial variables of type Location can be used with all visual stimulus types: pictures, movies, text, self-paced reading, and rapid serial visual presentation (RSVP). Once created, you apply it to your stimulus in the Event Editor’s **Stimulus** tab:

- Click on the **Settings** button; the Picture Settings dialog appears
- In the **Position** tab, click on the popup menu and change the selected option from **Fixed at location** to **Variable**





You will then see the Location trial variable that you have created. You can apply Location trial variables to the other stimulus types the same way.

## VARYING TEXT COLOR

Trial variables of type Color can be used with all textual stimulus types: text, self-paced reading, and rapid serial visual presentation (RSVP). Once created, you apply it to your stimulus in the Event Editor's **Stimulus** tab:

- Click on the **Settings** button; the Text Settings dialog appears
- In the **Text Color** tab, click on the popup menu and change the font color option from **Fixed** to **Variable**

You will then see the Color trial variable that you have created. You can apply Color trial variables to the other stimulus types the same way.

## VARYING BY TIME LIMIT

Trial variables of type Time Limit can be used with all events. You apply it to your stimulus in the Event Editor's **Input** tab:

- Enable the option to end an event on a time limit

- Click on the popup menu and change the time limit option from **Fixed** to **Variable**

End This Event and Move to the Next One:

☐ Immediately after the event is presented

☒ After any response from the participant

☐ After a correct response from the participant

☒ Or a time limit, whichever happens first. The time limit is:

Fixed:

☒ Variable:

☐ Delay feedback actions until time limit elapses

## MANUAL TRIAL LEVEL SELECTION

There are cases where random or sequential use of trial levels are not sufficient. Examples are threshold or staircase experiments where, for example, you want to increase or decrease exposure time based on the participant's reaction time. You need to use Manual level selection; when editing a trial variable, you can find this option in the **Level Selection** tab:

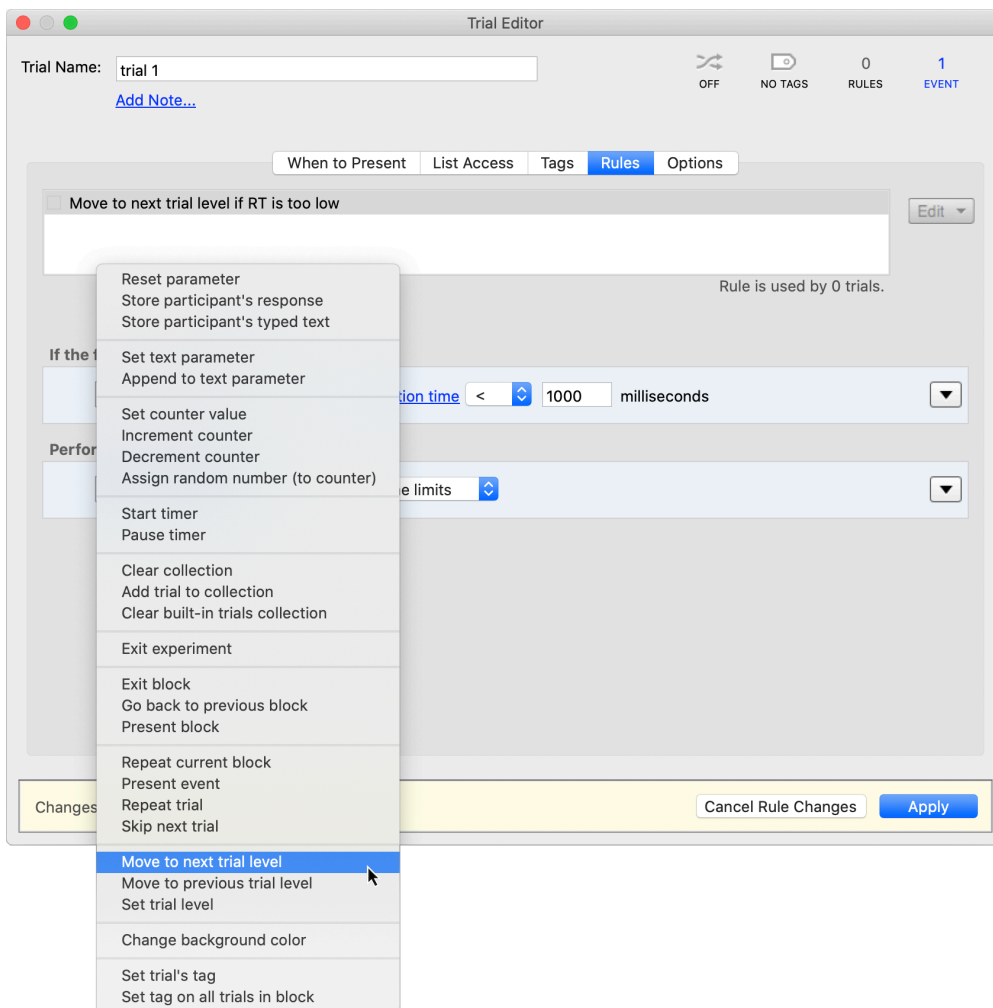
Edit Trial Variable

Name:  of type

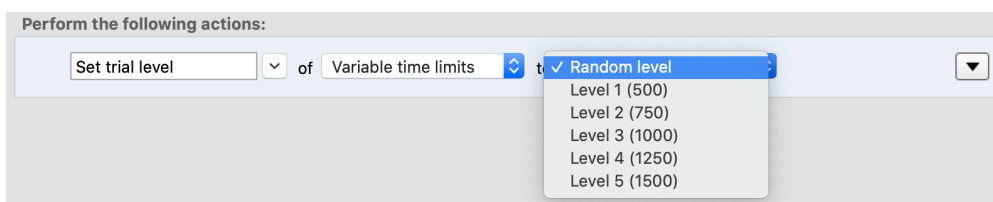
When choosing levels, use the following order:

Level selection is under your control: use Rules or feedback actions to pick a level or to move current level to the next or previous one.

If you do not do anything else at this point, the *current* trial level is the first one that you created, and SuperLab will use it over and over again. You must use an action to change the current level. This can be done in the Event Editor's **Feedback** tab, or in the **Rules** tab of either the Trial Editor or Block Editor, as shown here:



There are three actions available for use with trial variables that use Manual selection. Two of them let you move to the next or previous level — very useful with thresholding experiments. The third one, “Set trial level”, lets you select a specific level or even a random one:



There are also times when you want to determine what trial level you are currently using. For example, if your trial level is already at the shortest exposure time and the participant’s

reaction time is as good as can be expected, chances are you want to stop the experiment. SuperLab provides a criterion to do so:

A screenshot of a software interface for setting criteria. At the top, it says "If the following criteria are met:". Below this, there is a row of three elements: a text box containing "Level 'Variable time limi", a dropdown menu currently showing "✓ is" with "is not" as an option, and another text box containing "Level 1 (500)". To the right of these is a small square button with a downward-pointing triangle.

Note that this criterion works with *all* selection modes; it is not limited only to Manual selection.

## Randomizing Per Group

As you try out the randomization options in SuperLab, you will notice that most of them offer an option to randomize on a per group basis. This refers to SuperLab's Participant Groups feature which you can access from the **Experiment** menu. This feature lets you specify which blocks are presented and the order that they are presented in. It can be used when you need to divide your population of participants into groups based on certain factors such as age or gender; or to test a dependent variable.

When you choose to randomize on a per group basis, SuperLab will create a random sequence for each group that you create. All participants within a given group will see the same random sequence, but participants in group A will see a different sequence than participants in group B.

# Running Experiments & Viewing Data

## Before You Run Experiments

It is important that you be aware of some features *prior* to running your experiments because it will be difficult or impossible to adjust your data after the fact. Other features are less critical, but being aware of them would be useful.

### RESETTING THE RT TIMER

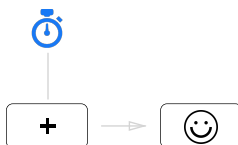
If you need to collect reaction time (RT) data, this is by far the most important feature to be aware of and can be found in the Event Editor's **Input** tab:

☐ **Reset RT timer** (all future reaction times will be measured relative to the onset of this event)  
(Note: SuperLab will always reset the timer on the onset of the first event in a trial.)

When you create a new event, the **Reset RT timer** option is not checked. When you run an experiment, SuperLab always resets the timer at the onset of the first event in the trial. Let's take the following simple trial design, a cue followed by the stimulus:



If the default **Reset RT timer** options are used, this will result in the following presentation:

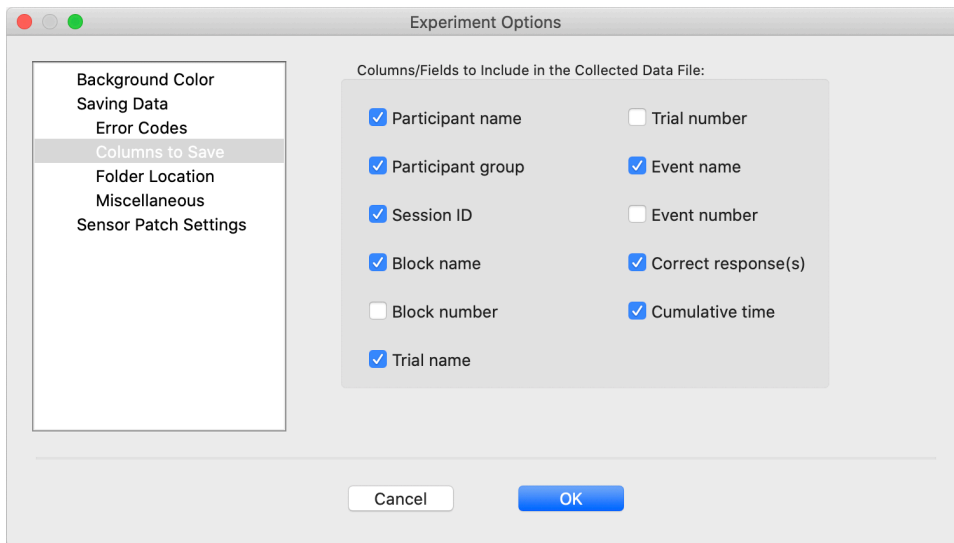


In a typical experiment, this is *not* desirable because the reaction times should be measured starting from when the face appears, not the cue. The **Reset RT timer** option for the Faces event must be checked.

## SAVED DATA

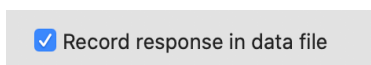
SuperLab saves the experiment's data in a tab-delimited text file, one file per participant. Every participant response occupies a row. There are options to choose which columns and which rows are saved.

You can access the column-saving options by clicking on the **Experiment** menu, choosing **Options...**, and then clicking on "Columns to Save":



Additional columns are also saved in the data file, one for each tag set, parameter, or trial variable.

You can also control which rows are saved by disabling the **Record response in data file** option in the Event Editor's **Input** tab:



This helps reduce clutter in the data file by eliminating unneeded rows, e.g. instructions. But be very careful: there is no way to recover this data after the experiment has been run.

## EXPERIMENT LOCK

Once you have completed building your experiment, it is a good idea to lock your changes so that no other changes can be made during the course of data collection. You can lock or unlock an experiment via the **Experiment** menu.

## Data Viewer

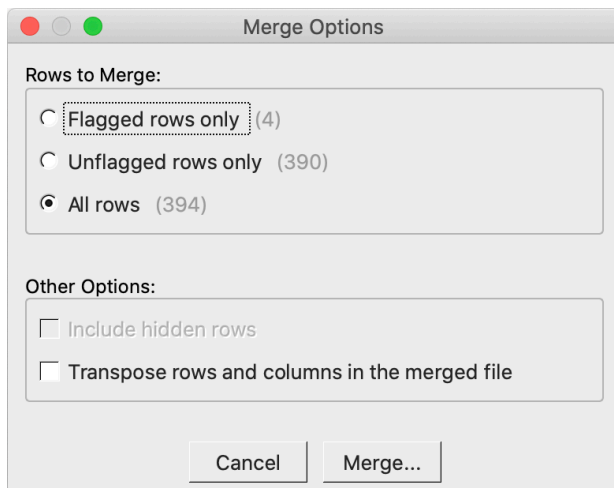
SuperLab's tab-delimited data files can be easily opened in Excel or other spreadsheet and statistical software. Another very convenient tool is the free Cedrus Data Viewer; you can download it from

<https://cedrus.com/superlab/dataviewer.htm>

Data Viewer is best thought of as a tool for cleaning up the data prior to moving on to statistical analysis. You can:

- Automatically flag rows that don't seem right, e.g. where the reaction time is more than two standard deviations from the mean
- Automatically flag records based on the correctness of the participant's response or the reaction time's range or limits
- View summary data such as counts, percentages, and average RTs – for one file or many files at the same time, and do so for each block, trial, tag set, or trial variable
- Combine any number of data files into a single merged file while optionally omitting flagged rows
- Hide records or columns
- Transpose records in the merged file

Do not hesitate to use these tools: the data files that you open in Data Viewer remain untouched. Changes will apply only when you click on the **Merge Selected Files...** button to export the files:



Basic usage of Data Viewer is illustrated on the [website](#), e.g. adding and viewing files, or looking at raw data vs. summary stats.



You can use the automatic flagging tools for 'drilling down' on data. After an initial set of records has been flagged based on one criterion, you can apply a second criterion on the remaining records.

You can use row coloring with Cedrus Data Viewer to make it easier to spot flagged records or incorrect responses. Row coloring is turned off by default but can be enabled by clicking on the **Rows** menu and selecting **Row Colors**:

Row Colors

**Colors to use when the response is:**

Correct: Don't use color      Self-Correct: Don't use color

Incorrect: Don't use color      No Response: Don't use color

Apply colors to:

☐ "Error Code" and "Reaction Time" fields only

☒ Entire row

**When row is flagged:**

☒ Override colors above, use Don't use color

- Don't use color
- Black
- Green
- Red
- Grey
- Other Color...

You can also move columns or hide them to suit your viewing preferences. You do so by right-clicking (or control-click on the Mac) on the column header; a menu will appear to offer the options.



# License Agreement

Nearly every software license sold is governed by a license agreement, sometimes called an End User License Agreement (EULA). Here is the EULA that governs the use of SuperLab, first in plain English.

## In Plain English

It's really simple:

1. If you have one license, no two persons may use SuperLab at any one time. If you have two licenses, no three persons can use SuperLab at the same time.
2. Whenever in doubt, refer to item 1 above.
3. SuperLab may not be installed on a virtual machine such as Parallels or VMware.

Our EULA offers more flexibility than other vendors', many of whom insist that the software may be installed on one machine only.

## Some Frequently Asked Questions

### **Can I develop an experiment on one computer and run it on others?**

No. You can run SuperLab to develop and collect data on only a single computer. If you want to collect data (but not edit or modify an experiment) on other computers, Cedrus offers the affordable SuperLab Remote.

### **Can I install SuperLab on the computer in the office and the one at home?**

Yes, provided there is no chance that another person may use it while you're using SuperLab at home.

## The Legal Text

### END-USER LICENSE AGREEMENT FOR CEDRUS SOFTWARE

**IMPORTANT** — This End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and the PUBLISHER from whom you acquired the software product SuperLab identified above ("SOFTWARE"). The SOFTWARE includes computer software, the associated media, any printed materials, and any "online" or electronic documentation. By installing, copying or otherwise using the SOFTWARE, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, Cedrus Corporation ("Cedrus") is unwilling to license the SOFTWARE to you. In such event,

you may not use or copy the SOFTWARE, and you should promptly contact Cedrus for instructions on return of the unused product(s) for a refund.

## Software LICENSE

The SOFTWARE is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE is licensed, not sold.

### 1. GRANT OF LICENSE.

This EULA grants you the following rights:

- **Software.** You may install and use one copy of the SOFTWARE on a computer.
- **Storage/Network Use.** You may also store or install a copy of the computer software portion of the SOFTWARE on the computer to allow your other computers to use the SOFTWARE over an internal network, and distribute the SOFTWARE to your other computers over an internal network. However, you must acquire and dedicate a license for the SOFTWARE for each computer on which the SOFTWARE is used or to which it is distributed. A license for the SOFTWARE may not be shared or used concurrently on different computers.
- **Backup Copy.** You may make a single backup copy of the SOFTWARE. You may use the backup copy solely for archival purposes.

### 2. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.

- **Limitations on Reverse Engineering, Decompilation and Disassembly.** You may not reverse engineer, decompile, or disassemble the SOFTWARE, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.
- **Separation of Components.** The SOFTWARE is licensed as a single product. Its component parts may not be separated for use on more than one computer.
- **Single COMPUTER.** The SOFTWARE is licensed as a single integrated product. The SOFTWARE may only be used with a single computer.
- **Software Transfer.** You may permanently transfer all of your rights under this EULA provided you retain no copies, you transfer all of the SOFTWARE (including all component parts, the media and printed materials, any upgrades, and this EULA), and the recipient agrees to the terms of this EULA. If the SOFTWARE is an upgrade, any transfer must include all prior versions of the SOFTWARE. Completing a Transfer of License form is required.
- **Termination.** Without prejudice to any other rights, Cedrus may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE and all of its component parts.
- **Virtual Machines.** SOFTWARE may not be installed and used on a virtual computer or any form of "guest" operating system or virtual "appliance".

### 3. UPGRADES.

If the SOFTWARE is an upgrade from another product, whether from Cedrus or another supplier, you may use or transfer the SOFTWARE only in conjunction with that upgraded product, unless you destroy the earlier version of SOFTWARE. If the SOFTWARE is an upgrade of a Cedrus product, you now may use that upgraded product only in accordance with this EULA. If the SOFTWARE is an upgrade of a component of a package of software programs which you licensed as a single product, the SOFTWARE may be used and transferred only as part of that single product package and may not be separated for use on more than one computer.

### 4. COPYRIGHT.

All title and copyrights in and to the SOFTWARE (including but not limited to any images, photographs, animations, video, audio, music, text and “applets,” incorporated into the SOFTWARE), the accompanying printed materials, and any copies of the SOFTWARE, are owned by Cedrus or its suppliers. You may not copy the printed materials accompanying the SOFTWARE. All rights not specifically granted under this EULA are reserved by Cedrus.

### 5. PRODUCT SUPPORT.

Product support for the SOFTWARE is provided by Cedrus or its resellers.

### 6. U.S. GOVERNMENT RESTRICTED RIGHTS.

The SOFTWARE and documentation are provided with RESTRICTED RIGHTS. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (2) of the Commercial Computer Software—Restricted Rights at 48 CFR 52.227-19, as applicable. Manufacturer is Cedrus Corporation / 1121 S. Meyler Street / San Pedro, CA 90731.

### 7. LIMITATION OF LIABILITY.

Under no circumstances and under no legal theory, tort, contract, or otherwise, shall Cedrus, its suppliers, or resellers be liable to you or any other person for any indirect, special, incidental, or consequential damages of any character including, without limitation, damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses. In no event will Cedrus be liable for any damages in excess of the amount Cedrus received from you for a license to the SOFTWARE, even if Cedrus shall have been informed of the possibility of such damages, or for any claim by any other party. This limitation of liability shall not apply to liability for death or personal injury to the extent applicable law prohibits such limitation. Furthermore, some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so this limitation and exclusion may not apply to you.

## 8. HIGH RISK ACTIVITIES.

The SOFTWARE and the output produced by the SOFTWARE is not fault-tolerant and is not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the SOFTWARE could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). Cedrus and its suppliers specifically disclaim any express or implied warranty of fitness for High Risk Activities.

## 9. WAIVER OF RESPONSIBILITY FOR MISUSE.

The SOFTWARE can process copyright material and has no way of knowing if such processing is in breach of copyright law. You agree that it is your responsibility to ensure that you only process material for which you own or have licensed the copyright and, further, to indemnify and hold harmless Cedrus from any claim whatsoever relating to your improper use of the SOFTWARE to process copyright materials.

## 10. MISCELLANEOUS.

This EULA represents the complete agreement concerning this license. The acceptance of any purchase order placed by you is expressly made conditional on your assent to the terms set forth herein, and not those in your purchase order. If any provision of this EULA is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. This EULA shall be governed by American law in the State of California (except for conflict of law provisions). The application the United Nations Convention of Contracts for the International Sale of Goods is expressly excluded.