13 Creating Experiments: Overview

The easiest way to start developing EyeLink experiments is to study the supplied templates included with the Experiment Builder software, installed at:

- Windows: "C:/Users/{User Name}/Documents/ExperimentBuilder Examples"
- macOS: "Documents\ExperimentBuilder Examples"

Each of these experiment templates illustrates a typical experimental paradigm – the examples are optimized for a screen resolution of 1920 x 1080 at 100% display scale. The following table provides a brief description of the experiments. A detailed analysis of each template's operations is documented in the following sections. More examples can be found in the Experiment Builder Examples discussion forum (https://www.sr-research.com/support/thread-349.html).

Experiment	Purpose
Simple	The basic experiment template, displaying a single word in the center of the screen in each trial. This example is used to introduce how to create an experiment with SR Research Experiment Builder step-by-step.
Stroop	The basic template for creating non-EyeLink experiments. This template illustrates the use of a results file, RT calculation, and audio feedback.
Picture	Displays an image on the screen
TextLine	Experiment to show a single line of text, illustrating the use of runtime interest area segmentation.
TextPage	Experiment to show a full screen of text using a multi-line text resource.
GCWindow	Demonstrates how to use real-time gaze position to display a gaze-contingent window.
Track	Displays the user's current gaze position during recording and illustrates how to set the resource position contingent on the current gaze position.
Change	Displays several almost identical screens rapidly, and illustrates use the of the fixation trigger.
Saccade	Illustrates the creation of a simple experiment for saccade/anti-saccade research.
Video	Illustrates creating an experiment display video clips using xvid codec.
Pursuit	Illustrates several kinds of sinusoidal movement in a pursuit task.
InfantVideo	Performs calibration and drift correction/check using animated calibration targets.
VisualWorld	Illustrates the implementation of the popular Visual World Paradigm. Audio is played through ASIO driver on Windows or OSX driver on macOS. Positions of the screen resources are set through an UPDATE_ATTRIBUTE action.
PosnerTask	Implements the classic Posner cueing task, in which either a neutral, valid or invalid cue is presented centrally prior to a target appearing in either a left or right placeholder. Participant is asked to indicate as rapidly as possible which side the target appeared on via a keypress response.
EEG Integration	This folder contains examples and instruction docs to illustrate integration with EEG systems from Biosemi, Brain Products, Neuroscan, and Magstim-EGI. All of the examples are based on the "Simple" and "Stroop" templates.

The discussion of the “simple” template should be read before working with any of other templates, as it illustrates most of the shared operations for all experiments. You may go over the “Stroop” example for creating non-eye tracking experiments. In general, we recommend reading through all of the templates before programming your own experiment. When creating your experiment, you may also refer to the Experiment Builder Project Checklist in Chapter 16.

Before making any changes to the existing examples, we suggest you first make a copy of the examples and then uncheck the "Read-only" box of the topmost experiment node.

14 Creating EyeLink Experiments: The First Example

To illustrate the use of Experiment Builder, we are going to create a very simple eye-tracking experiment that runs 12 trials. In each trial, a single word is displayed in the center of the screen (much like the "SIMPLE" template of the EyeLink C Programming API).

Creating an Experiment with SR Research Experiment Builder, consists of the following three overall steps:

- Create an Experiment
- Build and test run the Experiment
- Deploy the Experiment

Deploying the experiment generates a set of files used to run the experiment for data collection without relying on the Experiment Builder application.

14.1 Creating the Experiment

This section provides a step-by-step tutorial to walk you through the basics of creating an experiment with SR Research Experiment Builder.

14.1.1 Creating a New Experiment Project

To open Experiment Builder, in Windows click Start → All Applications → SR Research and choose "Experiment Builder". On macOS, go to the "Applications/Experiment Builder/" folder, and open the "ExperimentBuilder" application.

When the application starts:

1) Click "File → New" on the application menu bar.
2) In the following "New Project" dialog box, enter "Simple" in the "Project Name" edit box.
3) Click the button on the right end of the "Project Location" to browse to the directory where the experiment project should be saved. If manually entering the "Project Location" field, please make sure that the intended directory already exists.
4) Leave the "Templates" field as "None". Make sure the "EyeLink Experiment" box is checked for an EyeLink experiment.
5) Select the eye tracker version from the dropdown menu (EyeLink I, EyeLink II, EyeLink 1000, EyeLink 1000 Plus, EyeLink Portable Duo, or EyeLink 3). Or select "Current" to allow any EyeLink eye tracker (See Figure 14-1).

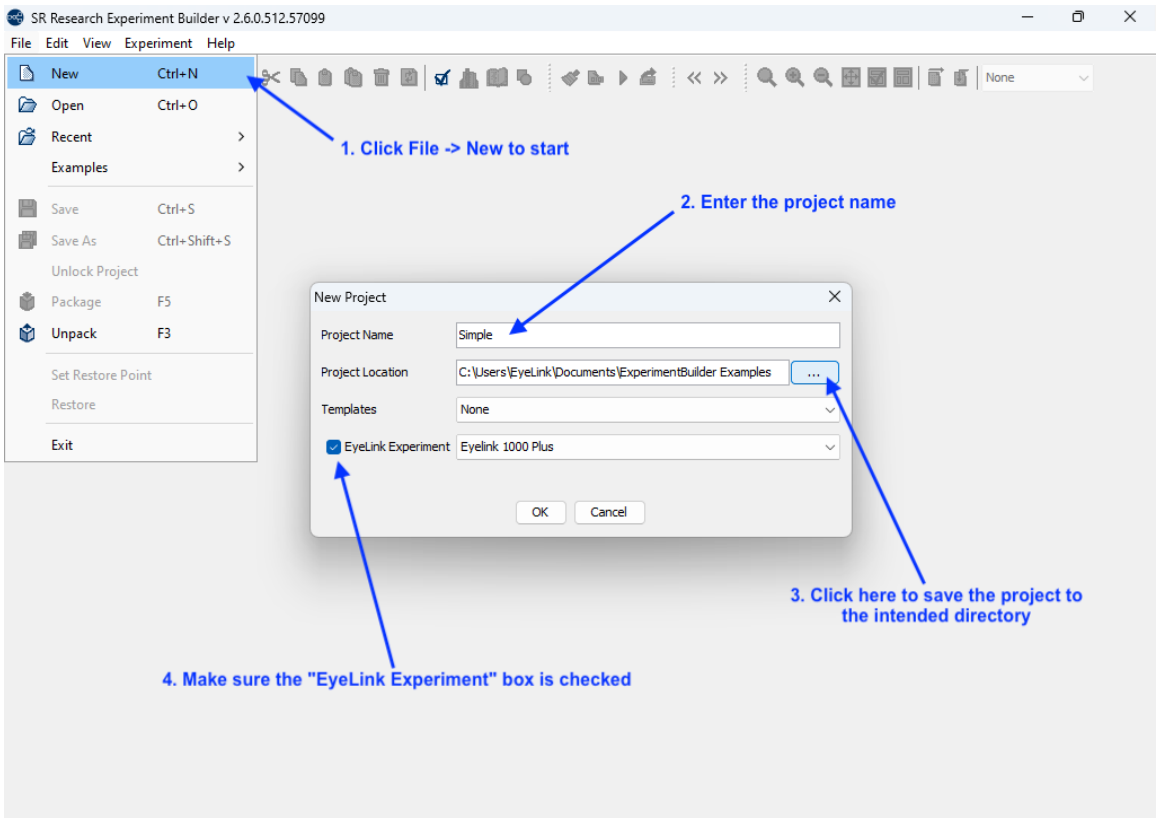

Figure 14-1. Creating a New Experiment Builder Project.

Important: Users should not manually add or remove files in the experiment directory. To maintain file integrity for the experiment projects created, any changes made to the experiment directory will be overwritten by Experiment Builder. Any image files, audio files, video files, etc., should be added or removed from the project through the Library Manager.

14.1.2 Configuring Experiment Preference Settings

After creating a new experiment session, check whether the default display and screen preference settings are appropriate for the experiment to be created.

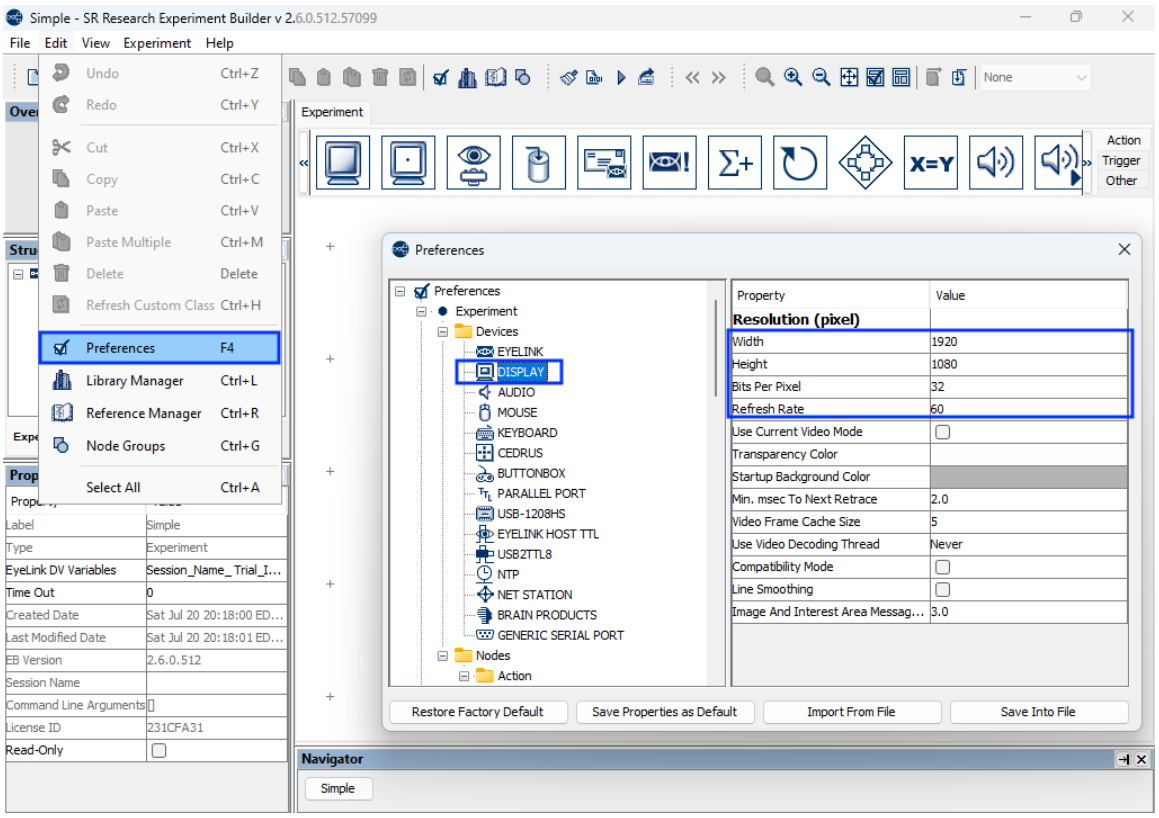

Figure 14-2. Configuring Preference Settings.

1) Select “Edit → Preferences” from the application menu bar or press the shortcut key “F4” on Windows. On macOS, click “ExperimentBuilder → Preferences” from the application menu bar or press Command ⌘+ “,”.
2) Click “Preferences → Experiment → Devices → Display” to check display settings. Make sure the settings (Width, Height, Bits per Pixel, and Refresh Rate) used in the current example are supported by your video card and monitor. Here the default values 1920 × 1080 × 32 × 60 Hz are used.
3) Click “Preferences → Screen” to check Screen Builder settings. Set the Location Type as "Center Position" and check the "Antialis Drawing" box.

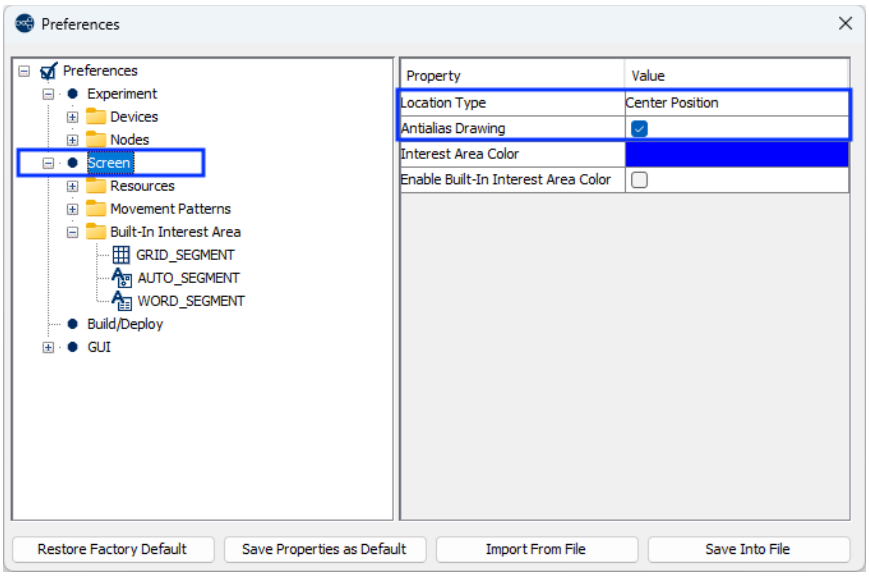

Figure 14-3. Setting the Screen Preferences.

4) **Tracker Version:** The default tracker version is set to “Current”. EyeLink I, II, 1000, 1000 Plus, Portable Duo, and EyeLink 3 users should also make sure the "Tracker Version" setting in the "Preferences -> Experiment -> Devices -> EyeLink" preferences is set to EyeLink I, EyeLink II, EyeLink 1000, EyeLink Portable Duo, or EyeLink 3. If you use EyeLink 1000, 1000 Plus, or Portable Duo, please make sure you configure the correct “Camera Mount” and “Mouse Usage setting.

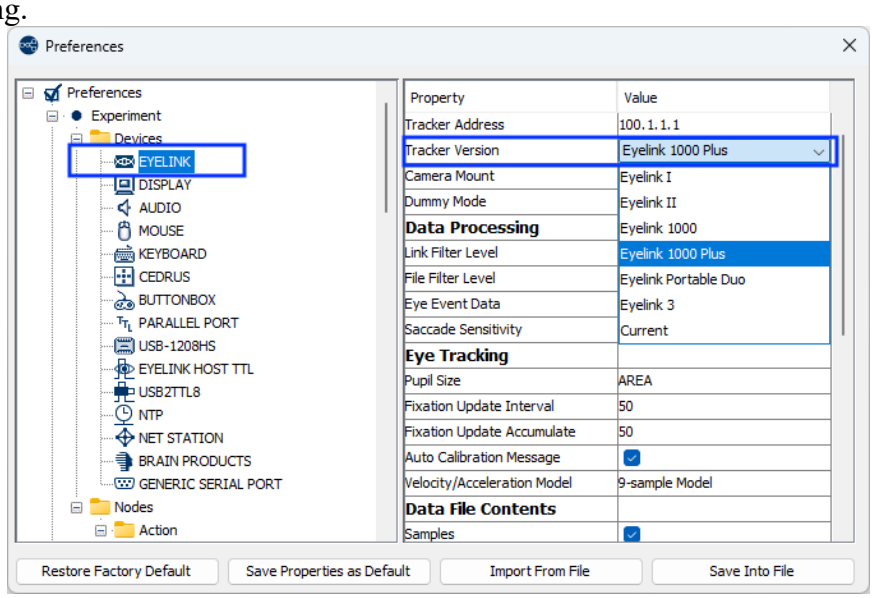

Figure 14-4. Setting the Tracker Version for the Experiment.

5) After changing any preference settings, if you would like to keep the new settings as defaults for all of your future experiments, click the "Save Properties as Default” button.
6) Once finished, press the close button on the dialog box.

If intending to use any characters that do not fit in the ASCII encoding range, including non-English characters (eg. à, è, ù, ç), special curved quotes, and any non-European language characters (e.g., Chinese characters), please also make sure the "Encode Files as UTF-8" box of the Build/Deploy node is checked (enabled by default in later versions of Experiment Builder; see Figure 14-5).

Failing to enable UTF-8 encoding when non-ASCII characters are used will result in the following build/run time warning:

> WARNING: warning:2001 You are using characters that ascii encoding cannot handle! Please change your encoding!

Likewise, if Chinese, Japanese, or Korean characters are used and UTF-8 encoding is not enabled, this will result in the following error:

> ERROR: error:2070 Internal Error. Could not create script. Please contact SR Research! Sorry: MemoryError: ().

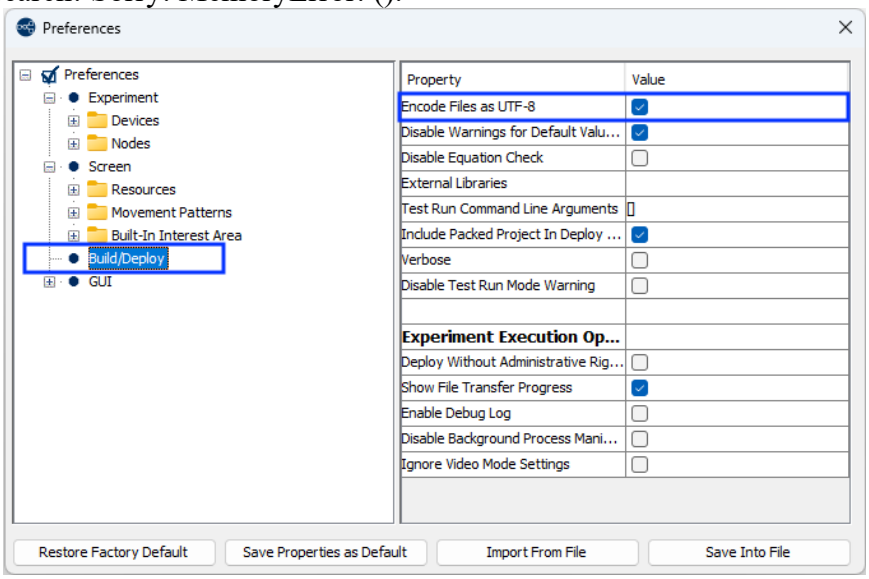

Figure 14-5. Setting the File Encoding for the Project.

14.1.3 The Topmost Experiment Layer

In this simple example, we start the experiment by displaying an instruction screen, perform a calibration, and then run 12 trials, with each trial showing a single word on the screen. We can start by adding the necessary nodes to the workspace of the topmost experiment layer:

1) Open the "Action" Tab of the component toolbox, then click and drag a "Display Screen" action into the work area.
2) Open the "Trigger" Tab of the component toolbox, then drag a "Keyboard" trigger into the work area.
3) Add an "EyeLink Button" trigger to the work space.
4) Add a "Timer" trigger to the work space.
5) Click the Timer trigger and set the duration to 120000 msec.

6) Open the “Action” Tab of the component toolbox and add a “Camera Setup” action to the work space.
7) Add a “Sequence” node to the work space. This will be our trial-level sequence.

Then we can continue by drawing the connections between the nodes in the BLOCK sequence:

8) Click and drag from the START node to the DISPLAY_SCREEN node.
9) Draw three connections from the DISPLAY_SCREEN action to the KEYBOARD, EL_BUTTON, and TIMER triggers. When a single action connects to several triggers, a number is added to each connection indicating the evaluation order among the three trigger types. In this experiment, it doesn’t matter which order the nodes are connected in.
10) Draw a connection from each of the three triggers to the EL_CAMERA_SETUP node, then from EL_CAMERA_SETUP to the SEQUENCE node.

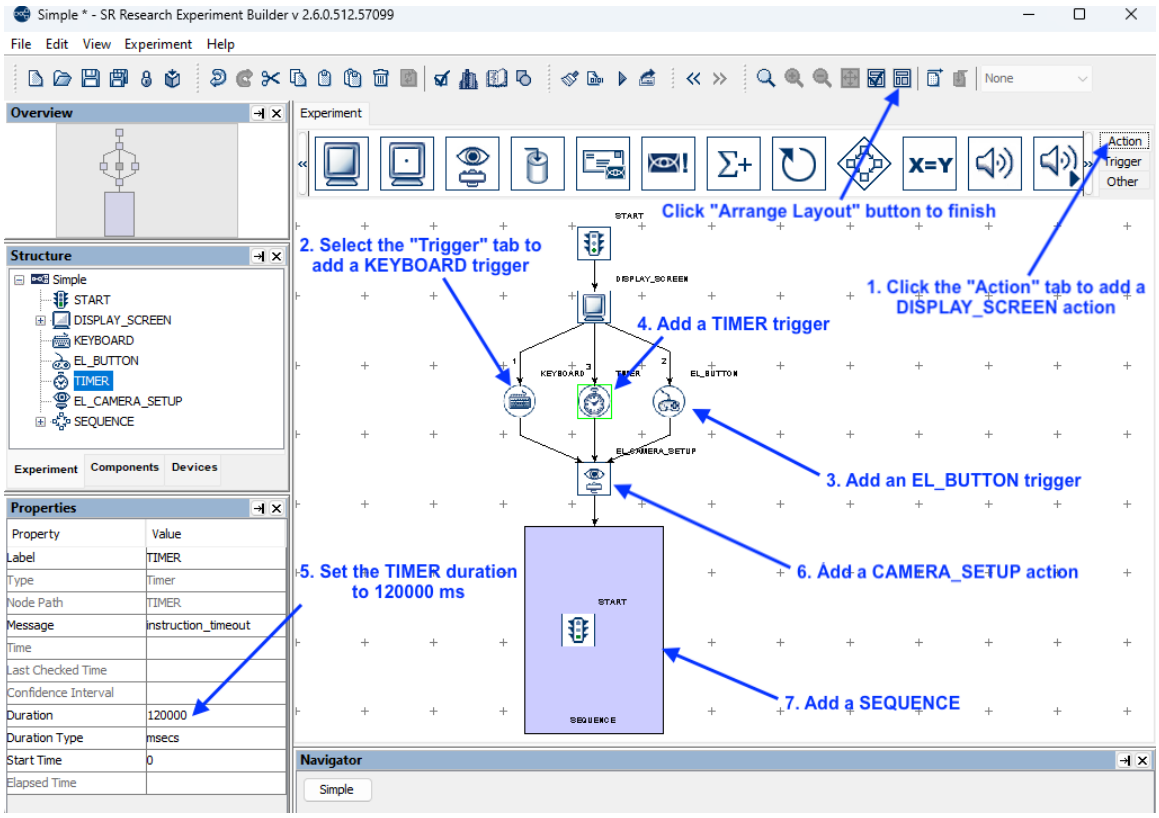

Figure 14-6. Adding Instruction to Block Sequence.

11) Right-click any blank area in the work window and select "Arrange Layout" in the popup menu or click the “Arrange Layout” button in the application toolbar.

14.1.4 Creating the Instructions Screen

Next, we can configure the DISPLAY_SCREEN node to display a set of instructions at the beginning of the experiment. In this example, we will create the instructions by adding a MultiLine Text Resource to the Display Screen; users may also create the instructions as an image file and use the Display Screen to display the image.

To start editing the screen, open the Screen Builder by double-clicking the DISPLAY_SCREEN node in the workspace.

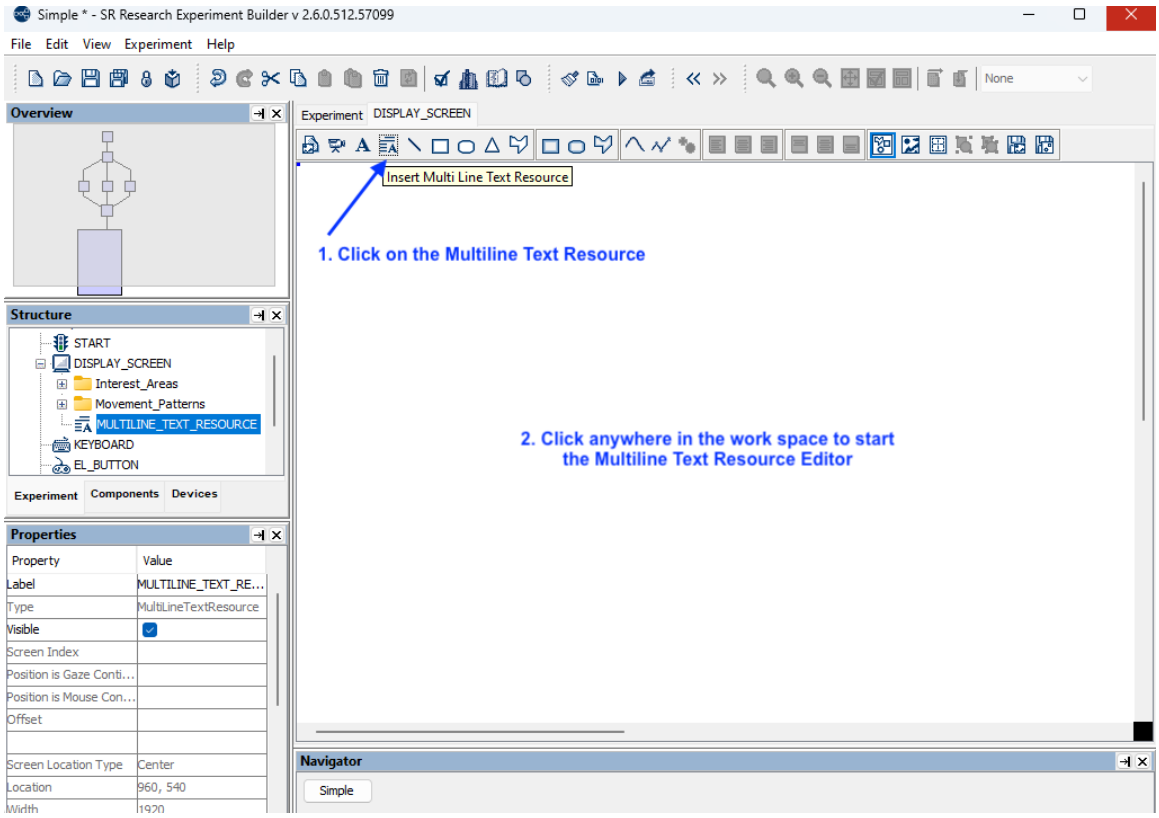

Figure 14-7. Adding Multiline Text Resource onto a Display Screen.

1) Once you've opened the Screen Builder, click the multiline text resource () button on the screen builder toolbar to select the resource to add.
2) Click anywhere on the screen to add the resource.

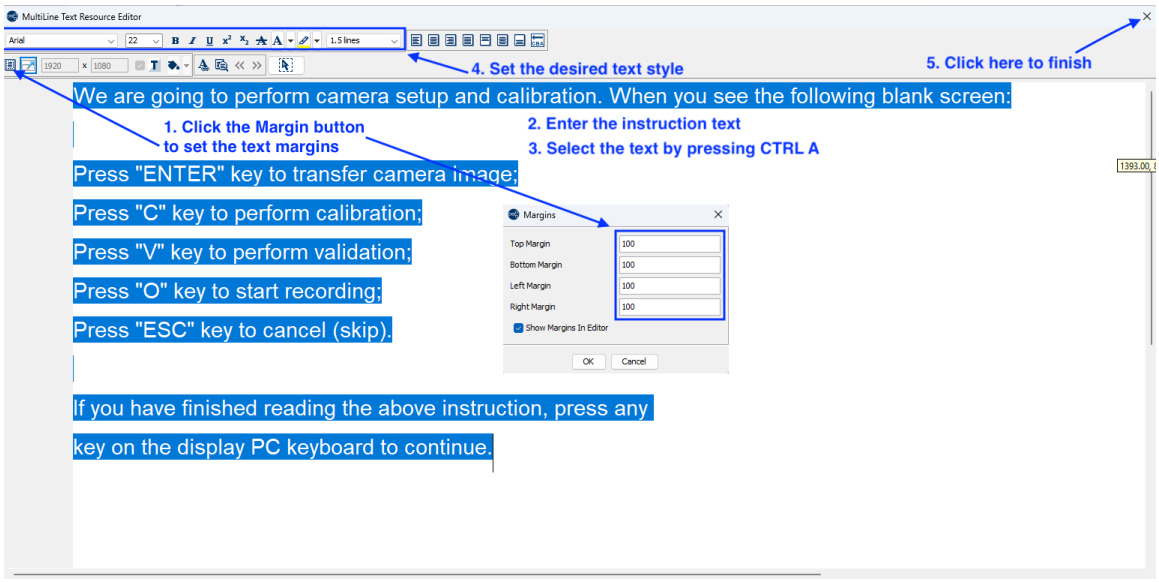

Figure 14-8. Create Instruction Screen.

In the following Multi-Line Text Resource Editor,

1) Click the “Margins” button in the toolbar to set the text margins. Enter 100 in all fields. Click the "OK" button on the dialog box.
2) Enter the desired instruction text.
3) Press Ctrl + A on Windows (Command ⌘ + A on macOS) to select all text entered.
4) Then click the buttons on the toolbar to set the desired text appearance (font name, font size, font style, alignment style, line spacing, and text color).
5) Click the “Close” button (✕) at the top right corner of the Multi-Line Text Resource Editor to finish.

14.1.5 Editing the Trial Sequence: Data Source

Next, we will design the Trial sequence, which will contain all the necessary triggers and actions in each trial. We will first create a data source to set the parameters in individual trials (see Figure 14-9). In this experiment, we will add two columns to the data source: “trial”, to serve as a trial identifier variable, and “word”, to determine the text displayed on each trial.

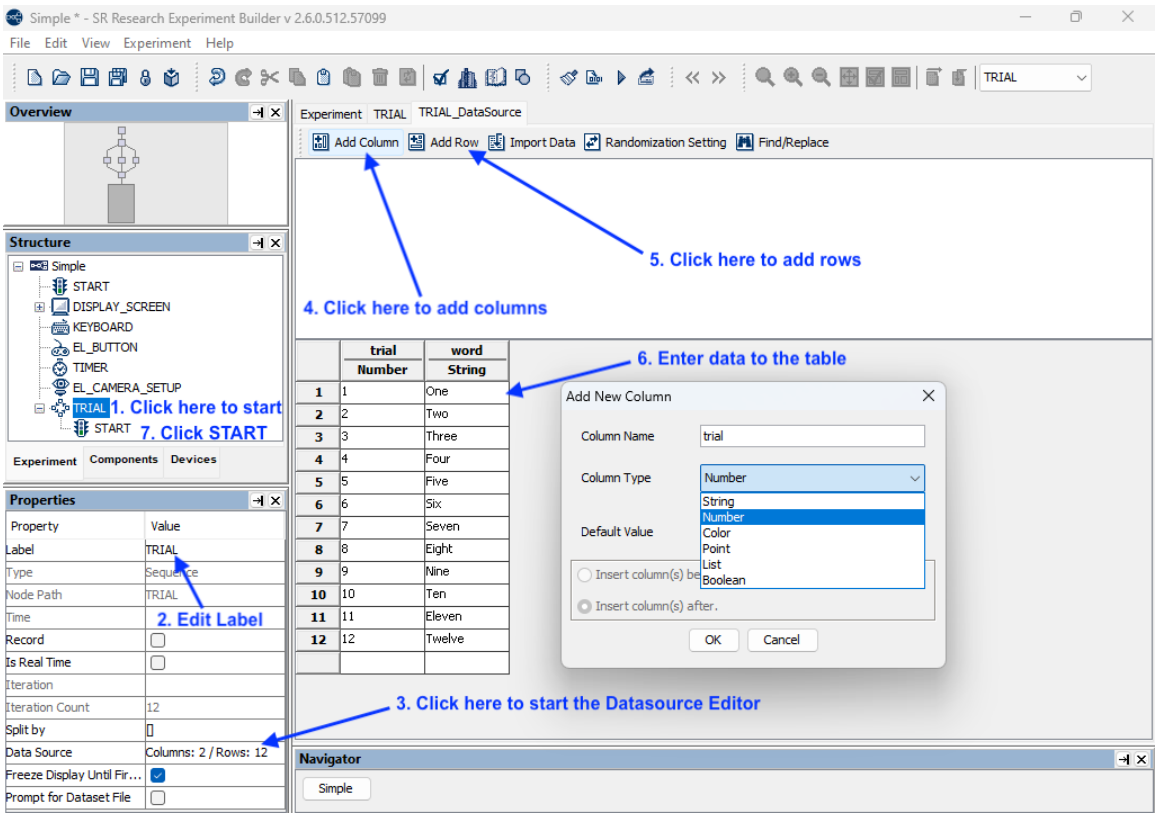

Figure 14-9. Creating Data Source.

1) First, select the last “SEQUENCE” node in the structure list.
2) In the Properties table, enter a new value for the Label, e.g., “TRIAL”.
3) Click the value cell of the “Data Source” property (where it says “Columns: 0 / Rows: 0”) to bring up the Data Source Editor.

Next we can create the Data Source columns and enter our values.

4) Click the "Add Column" button. Type "trial" (without quotes) in the Column Name box, and set the Column Type to "Number". Click the "OK" button to finish. Click the "Add Column" button again. Enter the Column Name as "word" and set Column Type to "String". Click "OK" to finish.
5) Click the “Add Row” button. Set the “Number of Rows” to 12 to add 12 rows of empty cells, then click “OK”.
6) To add a value to the Data Source, simply click in one of the empty cells, type the value into the editor, then press the Enter key. Set the values of the “trial” column as 1, 2, 3, … 12, and in the “word” column, enter the words “One”, “Two”, “Three”, etc., all the way to “Twelve”.
7) To enter the TRIAL sequence, double-click the TRIAL sequence node in the project workspace.

14.1.6 Editing the Trial Sequence: Preparing Sequence and Drift Correction

Next, we can start filling the contents of our TRIAL sequence.

Each trial should begin with a Prepare Sequence action, followed by a Drift Correct action, and then by the actual trial recording sequence (see Figure 14-10). The Prepare Sequence action preloads any image files or audio clips in the trial for real-time image drawing or sound playing, draws feedback graphics on the Host PC to evaluate participants' performance, and reinitializes trigger settings. Users should typically call the Prepare Sequence action before performing a drift correction.

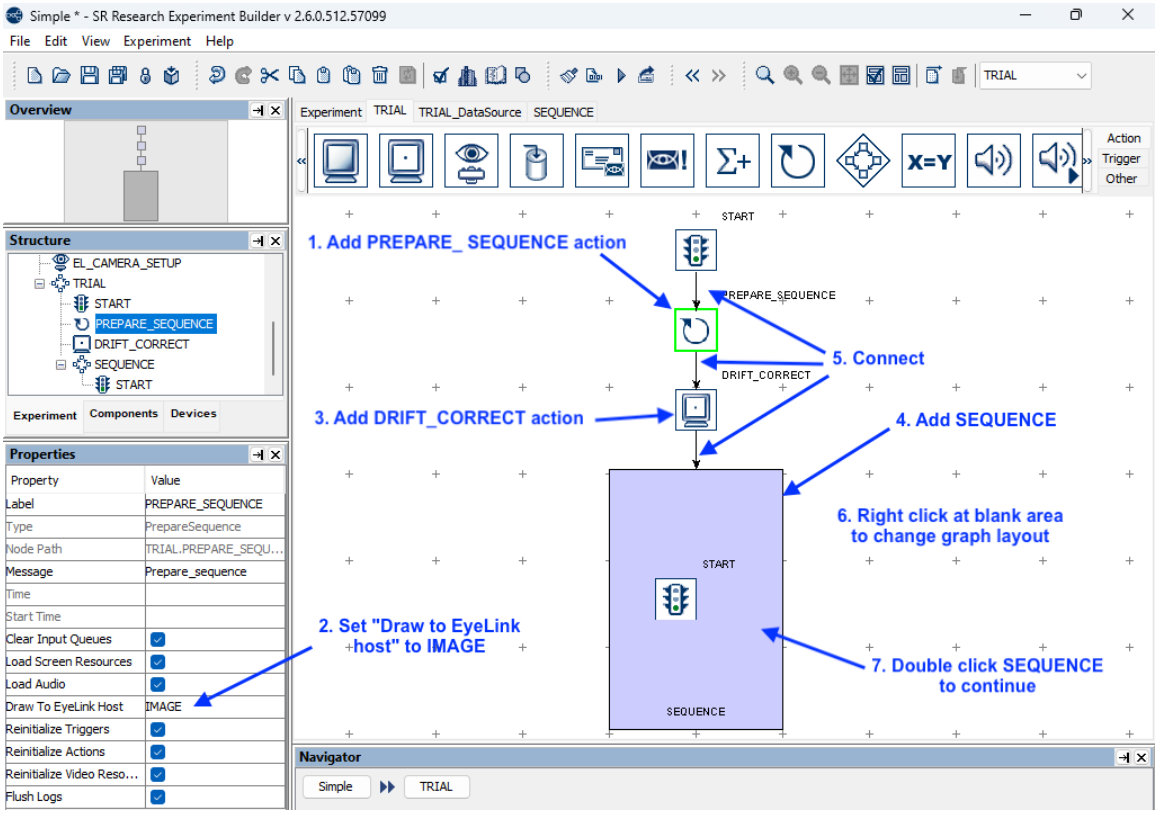

Figure 14-10. Editing Trial Sequence.

1) Open the "Action" Tab of the component toolbox, then drag a "Prepare Sequence" action into the workspace.
2) Select the added PREPARE SEQUENCE action and review the settings in the properties table. To draw an image from the trial or simple feedback graphics on the host screen to evaluate the participants' gaze position during recording, make sure the "Draw To EyeLink Host" field is set to "IMAGE" or "PRIMITIVE".
3) Drag a "Drift Correction" action into the workspace.
4) Then drag a "Sequence" node into the workspace.
5) Click and drag to draw a connection from the "START" node to "PREPARE_SEQUENCE", then from "PREPARE_SEQUENCE" to "DRIFT_CORRECTION", then from "DRIFT_CORRECT" to the "SEQUENCE" node.
6) Right-click any blank area in the work window and select "Arrange Layout" in the popup menu.

14.1.7 Editing the Recording Sequence

We will start by setting the properties of the recording sequence, as well as the actual contents of the trial recording (see Figure 14-11). In this simple recording sequence, we will display a screen and then wait for a button press response from the participant. The trial times out automatically if no response is made within 10 seconds. After the response or time out, the display screen is cleared. It is critical within the Recording sequence to fill in the "Message" field of action and trigger nodes corresponding to stimulus presentation and participant response so the timing of these events will be marked in the EDF.

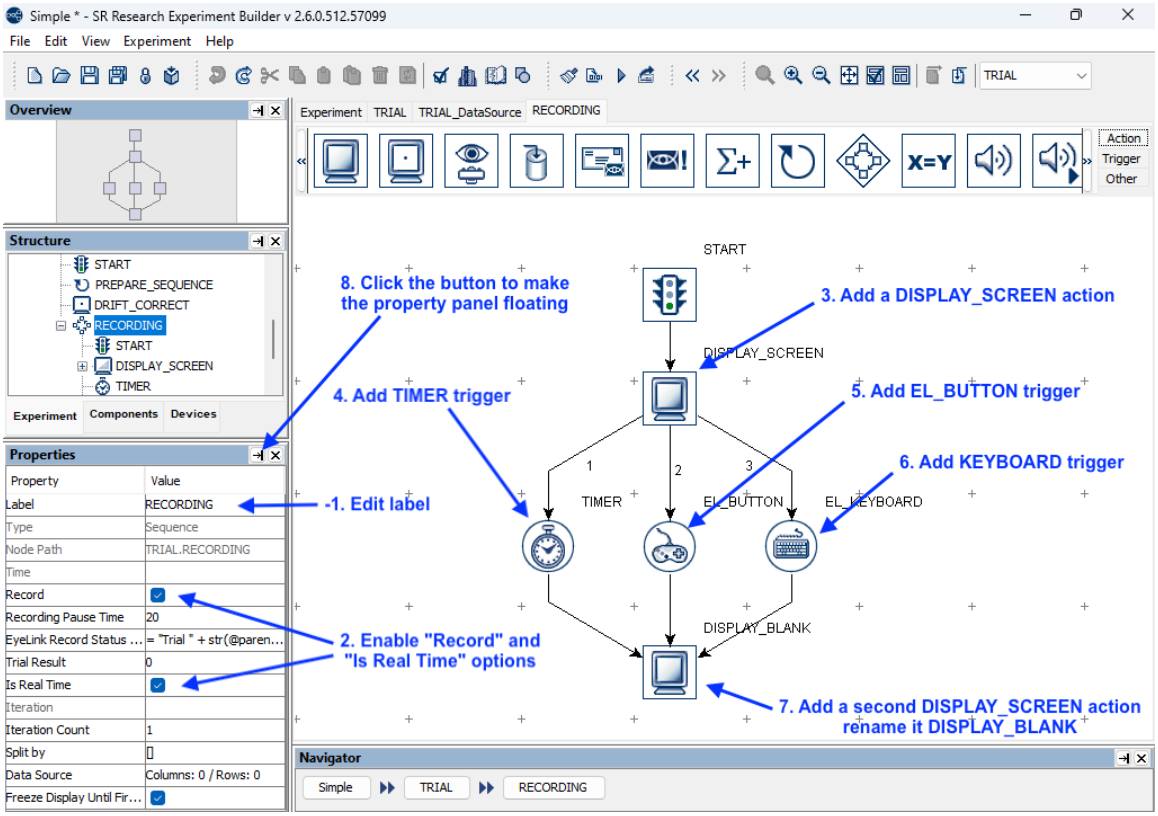

Figure 14-11. Editing Recording Sequence.

1) Select the newly added "Sequence" node and enter a new value for the Label, e.g., "RECORDING".
2) Make sure that the "Record" and "Is Real Time" checkboxes are checked. Double-click the "RECORDING" node in the structure list until seeing a "START" node under it. As we double click the "START" node, the content of the work area window is also updated.
3) Open the "Action" Tab of the component toolbox and drag a "Display Screen" action into the work area.
4) Then open the "Triggers" tab and drag a "Timer" trigger into the work space. Select the Timer object. In the "Duration" field enter 10000, and in the "Message" field, enter "Time out" (without quotes).
5) Drag an "EyeLink Button" trigger into the workspace.

6) Open the “Action” tab and drag a second “Display Screen” action into the workspace. Select the second Display Screen action, then set its Label, e.g., as “DISPLAY_BLANK”, and uncheck the “Send EyeLink DV Messages” box.
7) Draw connections from the “START” node to “DISPLAY_SCREEN”, from “DISPLAY_SCREEN” to both “TIMER” and “EL_BUTTON”, and from both “TIMER” and “EL_BUTTON” to “DISPLAY_BLANK”.
8) Right-click any blank area in the work space and select “Arrange Layout” in the popup menu to re-arrange the nodes.
9) Click the button in the properties window to make it a free-floating window.

14.1.8 Modifying the Properties of a Display Screen

We can next configure the property settings of the display screen actions in the Recording sequence (see Figure 14-12). For Data Viewer integration and for reaction time calculation, we can set a message to be written to the EDF file to indicate the time when the stimulus was visible to the participant. We can also configure which screen’s graphics are drawn the host screen so that the participants’ gaze position can be evaluated during recording.

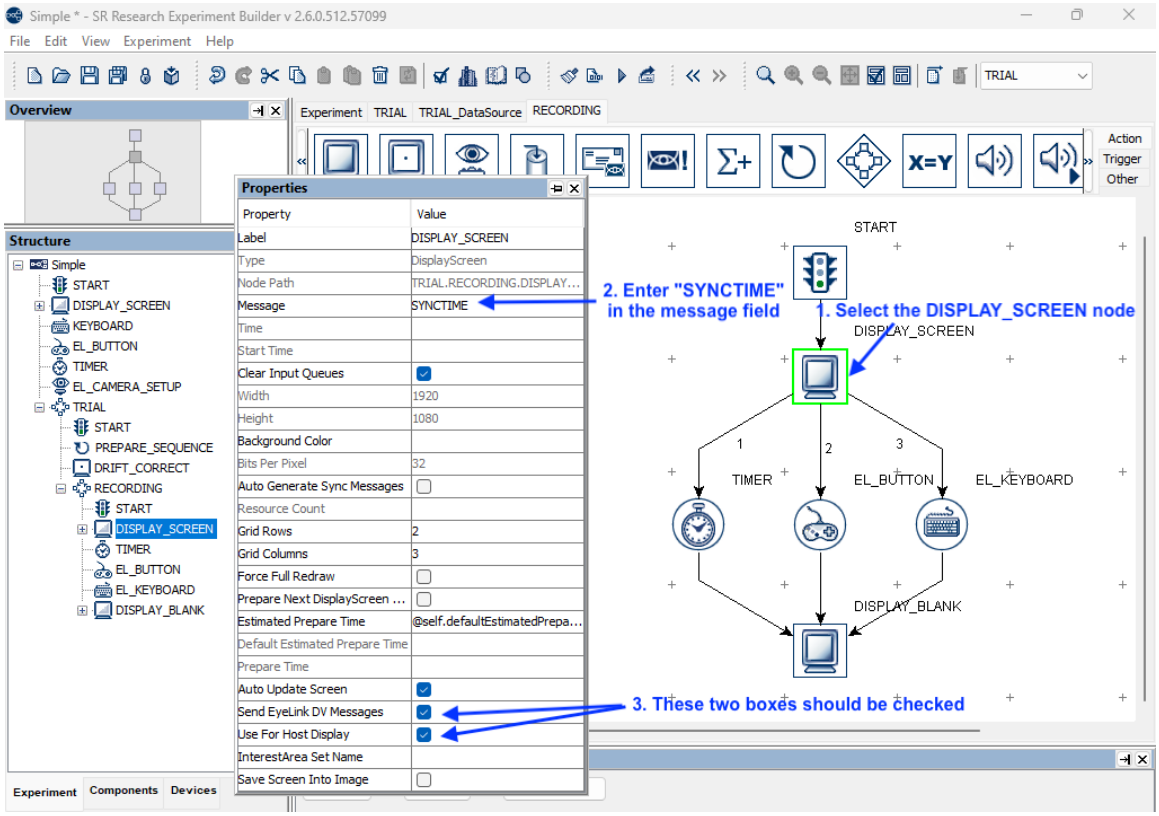

Figure 14-12. Modifying the Properties of DISPLAY_SCREEN Action.

1) Select the first DISPLAY_SCREEN node.
2) In the properties window, double-click the value field of the “Message” property. Type in a message, e.g., “SYNCTIME”, to mark the screen presentation, then press the Enter key.

3) Make sure the “Send EyeLink DV Messages” and “Use for Host Display” properties are checked.
4) Next, select the “DISPLAY_BLANK” action. Double click the value field of the “Message” property Type in a message, e.g., “blank_screen”, and then press the Enter key.
5) Make sure both the “Send EyeLink DV Messages” and the “Use for Host Display” checkboxes for the “DISPLAY_BLANK” action are unchecked.

14.1.9 Creating the Display Screen

We will next add a text resource to the display screen and modify the properties of the text resource, such as the text to be displayed, font name, size, and alignment style. We will also create an interest area for the text (see Figure 14-13). To open the Screen Builder window, double-click the “DISPLAY_SCREEN” object in the work space (***not in the structure list!***).

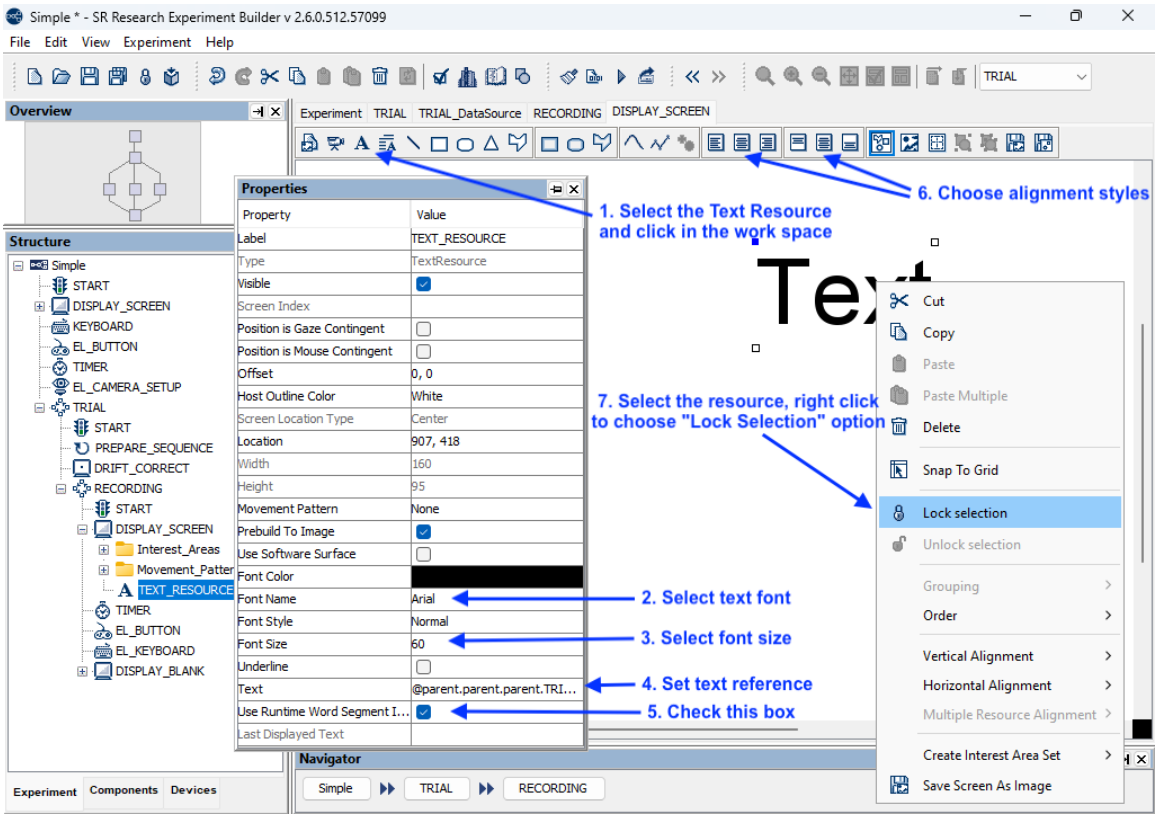

Figure 14-13. Adding Text to Display Screen.

1) Click the “Insert Text Resource” button () on the Screen Builder tool bar, then click any position in the screen area to add the resource.
2) Double click the Value of the Font Name property (e.g., “Times New Roman”). Select the desired font from the dropdown list—we’ll be using “Arial”.
3) Double click the value of Font Size (20). Enter the desired text size (40) in the text editor.

4) To set the text resource to load the text from the data source, click the value field of the "Text" property once, then click the […] button to open the Attribute Editor dialog (see Figure 14-14).
    a. Click the DataSource node under the "TRIAL" sequence in the Node Selection list.
    b. Double click the "word" attribute in the Node Attributes panel. This will update the contents of the "Attribute" panel as "@parent.parent.parent.TRIAL_DataSource.word@".
    c. Click the "OK" button to finish.

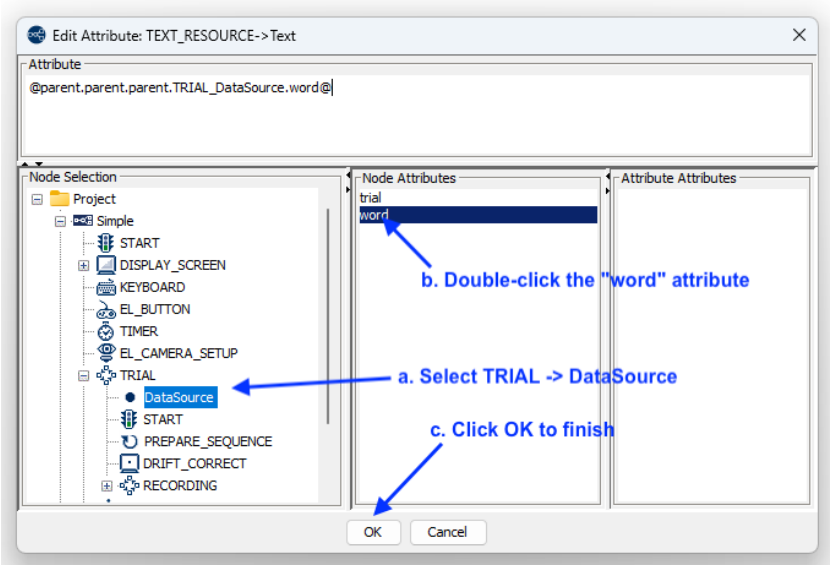

Figure 14-14. Showing Text by Referring to Data Source.

5) Check the "Use Runtime Word Segment" box. This will create interest areas automatically for the text used.
6) Select the newly added text resource, click both the "Horizontal Center Alignment" and "Vertical Center Alignment" ( ) buttons to place the text in the center of the screen.
7) Select the text resource in the work area, then right-click the resource and select the "Lock Selection" option so that the resource will not be moved accidentally.

14.1.10 Writing Trial Condition Variables to EDF file

Users may configure which variables, including variable nodes and data source columns, should be written to the EDF file so the experimental conditions of each trial can be identified during analysis (see Figure 14-15). In Experiment Builder 2.0, all newly-added variable nodes and data source columns will automatically be added to the list of trial variables.

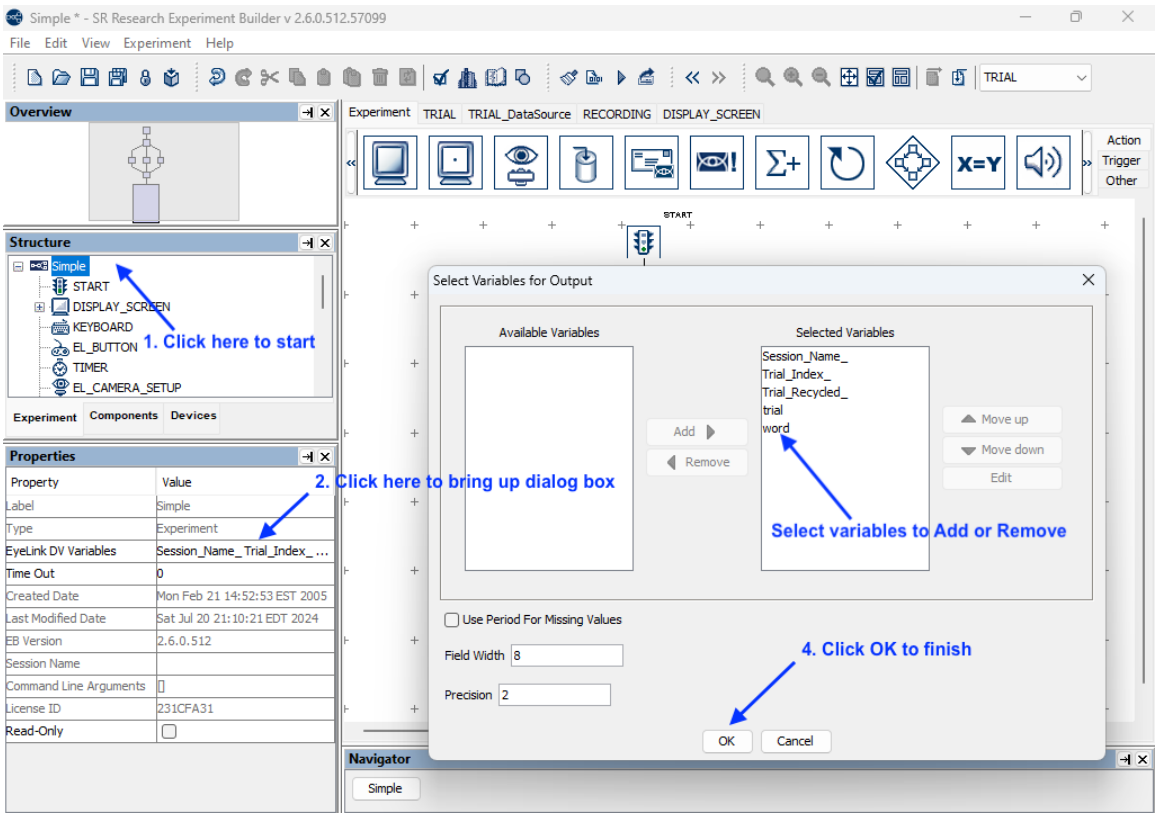

Figure 14-15. Configuring the EyeLink DV Variables.

1) Click the Experiment node in the structure list (the topmost node in the tree).
2) In the properties table, click the value field of the "EyeLink DV Variables" property.
3) In the following dialog box, the "Selected Variables" panel on the right lists all the data source columns and variables currently selected to be written to the EDF as trial variables, and "Available Variables" will show any variables not selected to be written as trial variables. To add or remove items from the Selected Variables, select the item(s) to be moved, then click the "Add" button ( ) or "Remove" button ( ). To configure the order of the variables in the output, select the item(s) to be moved and click the "Move Up" ( ) or "Move Down" ( ) buttons. In this example, make sure both the "Trial" and "Word" columns are included in the Selected Variables list.
4) Click "OK" to finish.

14.1.11 Showing Experiment Progress Message on Tracker Screen

During trial recording, a text message can be displayed at the bottom of the tracker screen so the experimenter can be informed of the experiment progress (see Figure 14-16). In this example, we will display a text message like "Trial 1/12 One" on the tracker screen, indicating the current trial and the word displayed.

1) Select the "RECORDING" sequence node in the structure list.

2) In the properties panel, click the value field of the “EyeLink Record Status Message” property once, then click the […] button to open the Attribute Editor dialog.
3) In the attribute editor, enter the following equation:

```
= "Trial " + str(@parent.iteration@) + "/ " + str(@parent.iterationCount@)
+ " " + str(@TRIAL_DataSource.word@)
```

4) Click the “OK” button to finish.

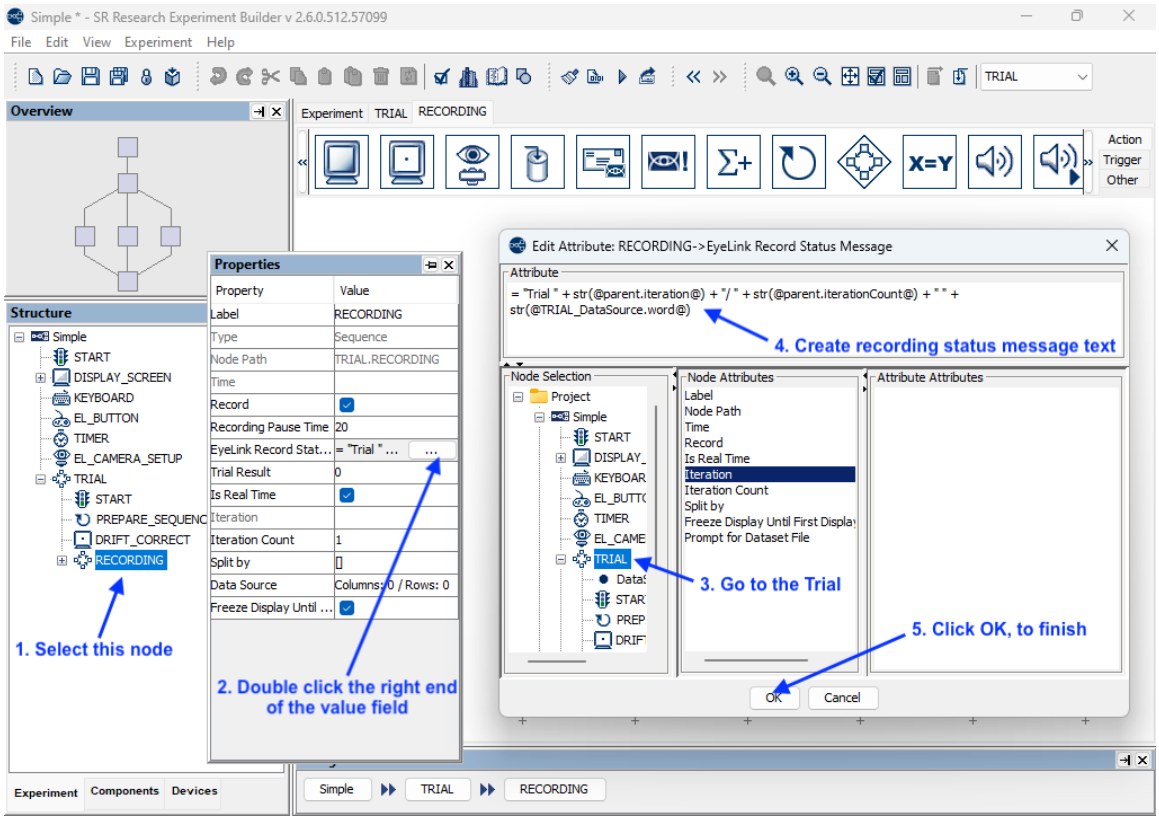

Figure 14-16. Creating Trial Recording Status Message.

14.2 Building the Experiment

Now your first experiment is created. If you haven’t saved your experiment project yet, click the Save ( ) button on the application tool bar. Click “Experiment → Build” to build the experiment. An “Output” tab will be opened in the Graph Editor Window and build information will be displayed. Watch for error messages (displayed in red) and warning messages (in brown) during building. If any error or warning messages appear, double-click on an error or warning message in the output tab to highlight the node or screen resource that produced the error/warning.

Once the project builds successfully, users may test run the experiment by clicking on “Experiment → Test Run” from the application menubar. This will try connecting the

Display PC to the Host PC to run the experiment code, so the Display PC must have an Ethernet connection to the Host PC (or Dummy Mode may be enabled). Please note that the "Test Run" should only be used for testing and debugging experiment code, and not for actual data collection. To collect experiment data, users should run the deployed version of the experiment (see the next section).

14.3 Deploying the Experiment

After the experiment is built, users must "deploy" the experiment into a new folder by clicking Experiment → Deploy (see Section 4.12). This will create an executable version of the experiment so it can run without relying on the Experiment Builder application. If a data source is used, this will create a "datasets" subdirectory with a copy of the data set file in it. If desired, users may create several data set files with the external randomizer application (see Section 9.6.2).

14.4 Running the Experiment

To run the experiment, first make sure the EyeLink host software is running, and the network connection between the host and display computers has been established. Then go to the directory where the experiment was deployed and click "simple.exe" to start the experiment. A dialog box will prompt for an EDF file name (the name should be no more than 8 characters, featuring only letters, numbers and the underscore "_" character). Click the "OK" button to continue. Following the initial welcome message, the participant will be shown the camera setup and calibration screen; the recording can be started following a calibration, validation, and pre-trial drift correction. After running all of the trials, an EDF file will be transferred to the display computer. It may take some time to complete the file transfer, so please be patient.

The following sections list some common errors while attempting to run an experiment.

14.4.1 Error in Initializing Graphics

If you start the experiment and see an "Could not initialize display to ***" error, please check whether the display settings (screen resolution, color bits, and refresh rate) specified for the experiment are supported by your video card and monitor (see Figure 14-17). If not, please set the correct display settings in "Preferences → Experiment → Devices → DISPLAY". This error can also be caused by an outdated graphics driver or lack of a proper driver – please make sure the graphics card driver is updated.

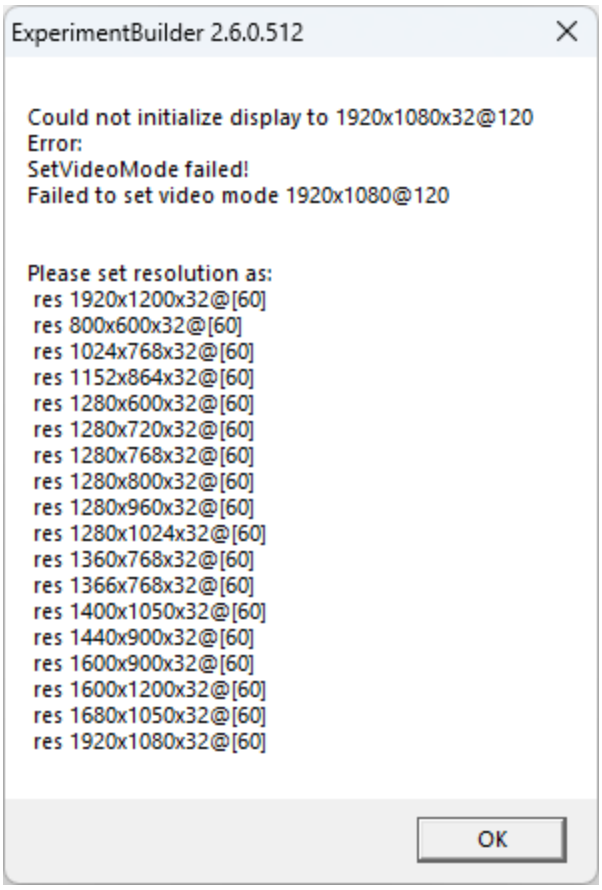

Figure 14-17. Error in Initializing Graphics.

14.4.2 Invalid Tracker Type

The latest version of SR Research Experiment Builder is compatible with EyeLink I, EyeLink II, EyeLink 1000, EyeLink 1000 Plus, EyeLink Portable Duo, and EyeLink 3 eye trackers. The default tracker version is set to EyeLink 1000 Plus (see "Preferences → Experiment → Devices → EYELINK"). If the eye tracker specified in the preferences doesn't match the eye tracker being used, Experiment Builder will display an error message like the one pictured in Figure 14-18—if you see this message, please set the correct tracker version in the device settings (see Figure 14-4).

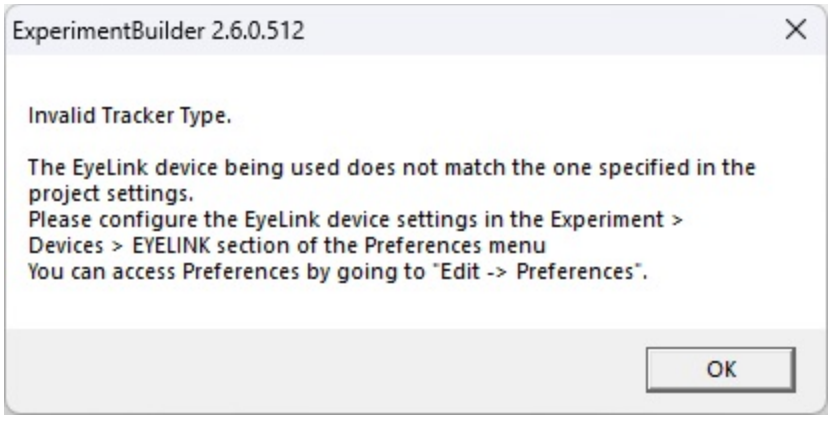

Figure 14-18. Error in Tracker Version.

15 Creating Non-EyeLink Experiments: Stroop Effect

This chapter illustrates using SR Research Experiment Builder to create a non-eye-tracking experiment. This Stroop sample experiment demonstrates testing the Stroop Effect with keyboard response: the participant is asked to respond to the colors of the words as quickly and as accurately as possible. For example, for the word "BLUE", the participant should respond "RED" instead of "BLUE".

15.1 Creating a New Experiment Project

To open Experiment Builder in Windows click Start → All Applications → SR Research and choose "Experiment Builder". On macOS, go to "Applications/ExperimentBuilder", then open the "ExperimentBuilder" application. When the application starts:

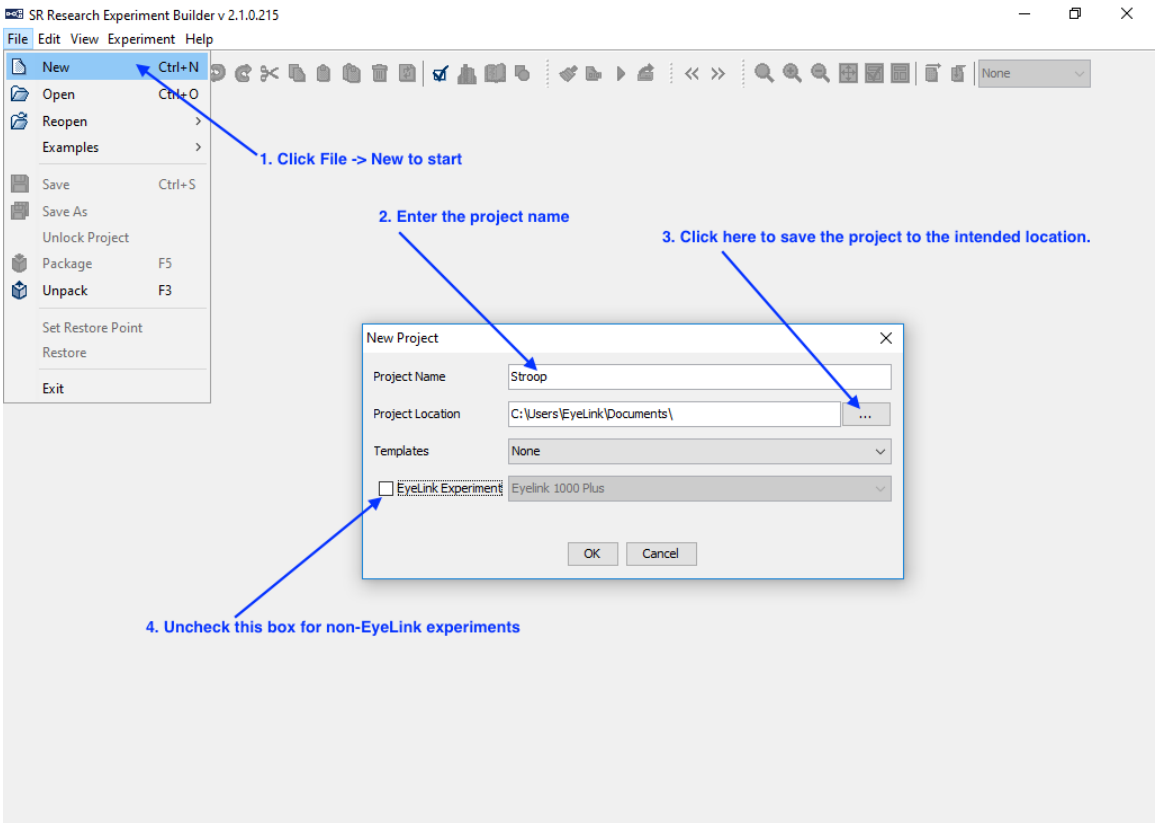

Figure 15-1. Creating a New Experiment Builder Session.

1) Click "File → New" on the application menu bar.
2) In the following "New Project" dialog box, enter "Stroop" in the "Project Name" edit box.
3) Click the button on the right end of the "Project Location" to browse to the directory where the experiment project should be saved. If manually entering the "Project Location" field, please make sure that the intended directory already exists.

4) Make sure the “EyeLink Experiment” box is unchecked for a non-EyeLink experiment.

15.2 Configuring Experiment Preference Settings

After creating a new experiment session, check whether the default display and screen preference settings are appropriate for the experiment to be created.

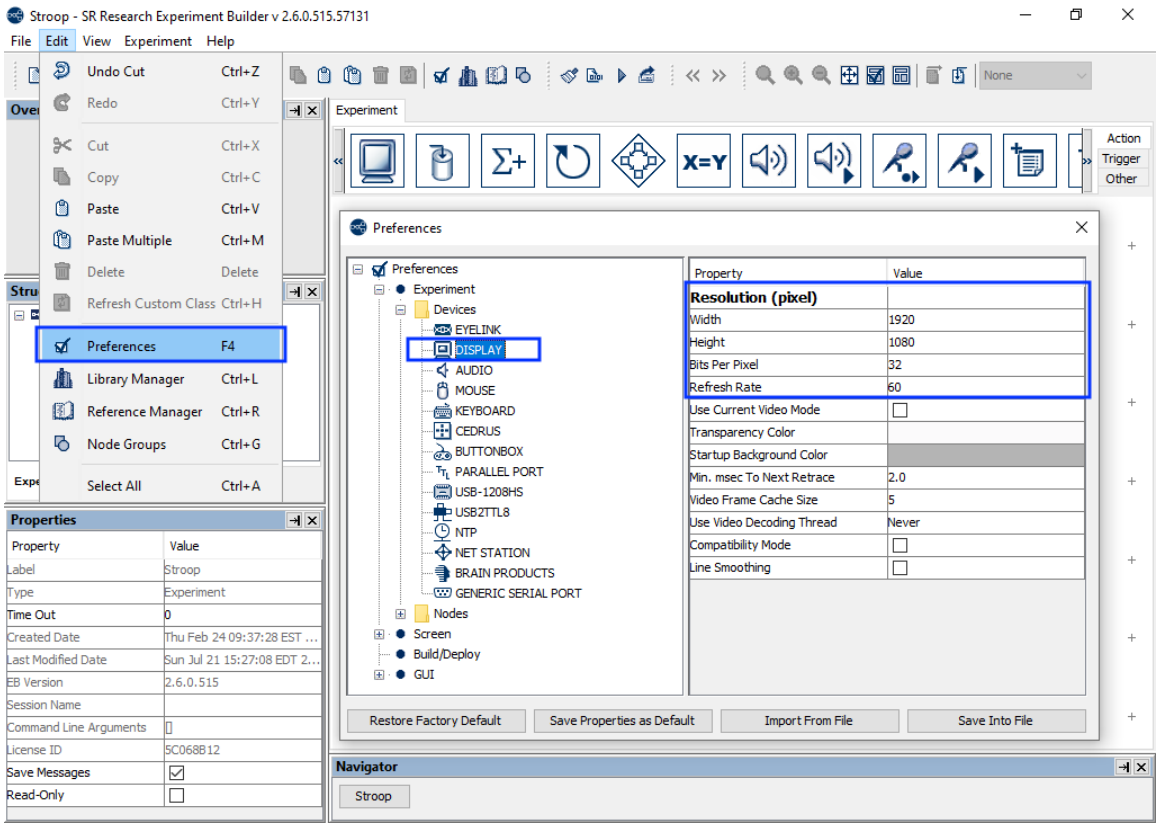

Figure 15-2. Editing Project Display Preferences.

1) Select “Edit → Preferences” from the application menu bar or press the shortcut key “F4” on Windows. On macOS, click “ExperimentBuilder → Preferences” from the application menu bar or press Command ⌘+ “,”.
2) Click “Preferences → Experiment → Devices → Display” to check display settings. Make sure the settings (Width, Height, Bits per Pixel, and Refresh Rate) used in the current example are supported by your video card and monitor.
3) Click “Preferences → Screen” to check Screen Builder settings. Set the Location Type as "Center Position" and check the "Antialis Drawing" box.

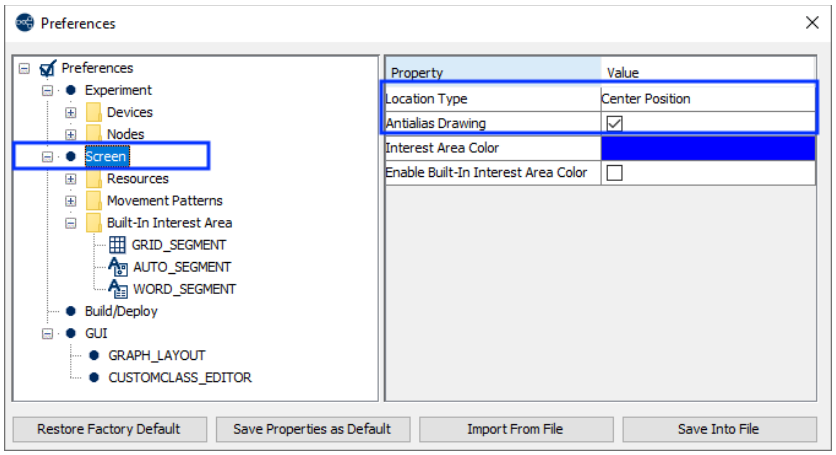

Figure 15-3. Setting the Screen Preferences.

4) After changing any preference settings, if you would like to keep the new settings as defaults for all of your future experiments, click the "Save Properties as Default" button.
5) Once finished, press the close button on the dialog box.

If intending to use any characters that do not fit in the ASCII encoding range, including non-English characters (eg. à, è, ù, ç), special curved quotes, and any non-European language characters (e.g., Chinese characters), please also make sure the "Encode Files as UTF-8" box of the Build/Deploy node is checked (enabled by default in later versions of Experiment Builder; see Figure 15-4).

Failing to enable UTF-8 encoding when non-ASCII characters are used will result in the following build/run time warning:

> WARNING: warning:2001 You are using characters that ascii encoding cannot handle! Please change your encoding!

Likewise, if Chinese, Japanese, or Korean characters are used and UTF-8 encoding is not enabled, this will result in the following error:

> ERROR: error:2070 Internal Error. Could not create script. Please contact SR Research! Sorry: MemoryError: ().

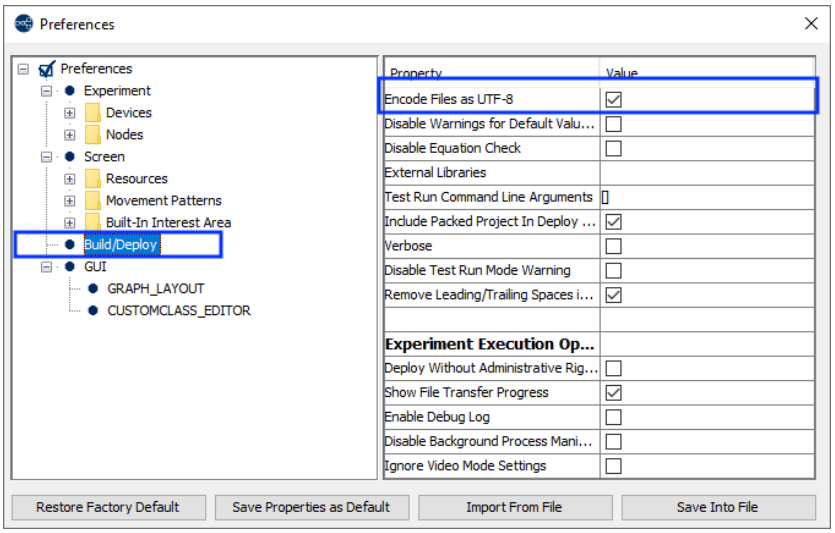

Figure 15-4. Editing Project Build/Deploy Preferences.

15.3 Creating the Experiment Block Sequence

In this example, we are going to run two blocks of nine trials each. The first step is to add a block sequence for repeating blocks (see Figure 15-5). Then we will add a results file to save data output.

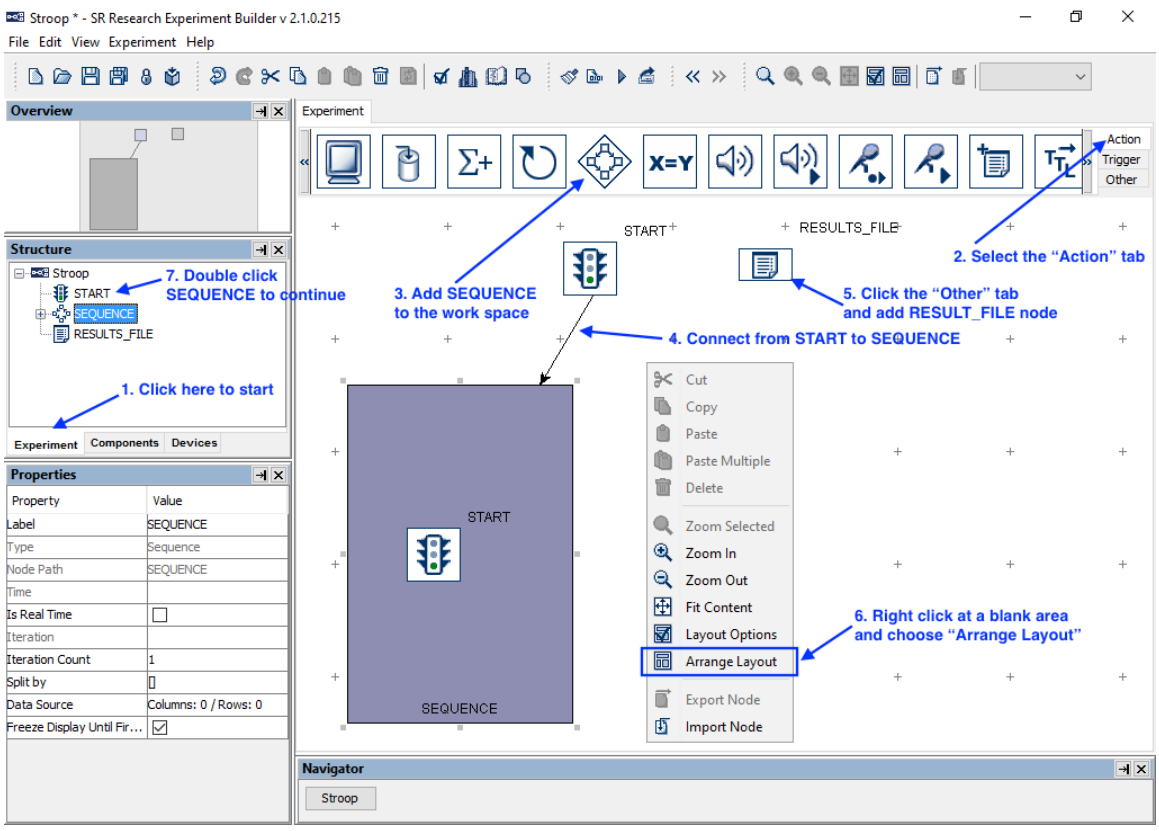

Figure 15-5. Creating Experiment Block Sequence.

1) Click the Experiment Tab in the Project Explorer Window to start.
2) Open the “Action” Tab of the component toolbox.
3) Click and drag the “Sequence” node from the component toolbox into the work space.
4) Connect the “START” and “SEQUENCE” nodes by clicking and dragging the mouse from the “START” node to the “SEQUENCE” node. (**Note:** If the “START” node is selected, clicking and dragging will move the node instead of drawing an arrow. To de-select the node, simply click in an empty area of the work space.)
5) Open the “Other” Tab of the component toolbox and add a “RESULTS_FILE” node to the graph.
6) To rearrange the nodes into an orderly layout onscreen, right-click any blank area in the work window and select "Arrange Layout" in the popup menu.
7) Select the SEQUENCE node in the structure list to continue.

15.4 Editing the Block Sequence

Next, we can edit the properties of the Block Sequence. We will first set the “Label” of the sequence to give it a meaningful name, then set the “Iteration Count” to the number of blocks to be tested (see Figure 15-6). (We do not edit the “Split by” field for the block-level sequence; we will instead set the “Split by” of the trial-level sequence.)

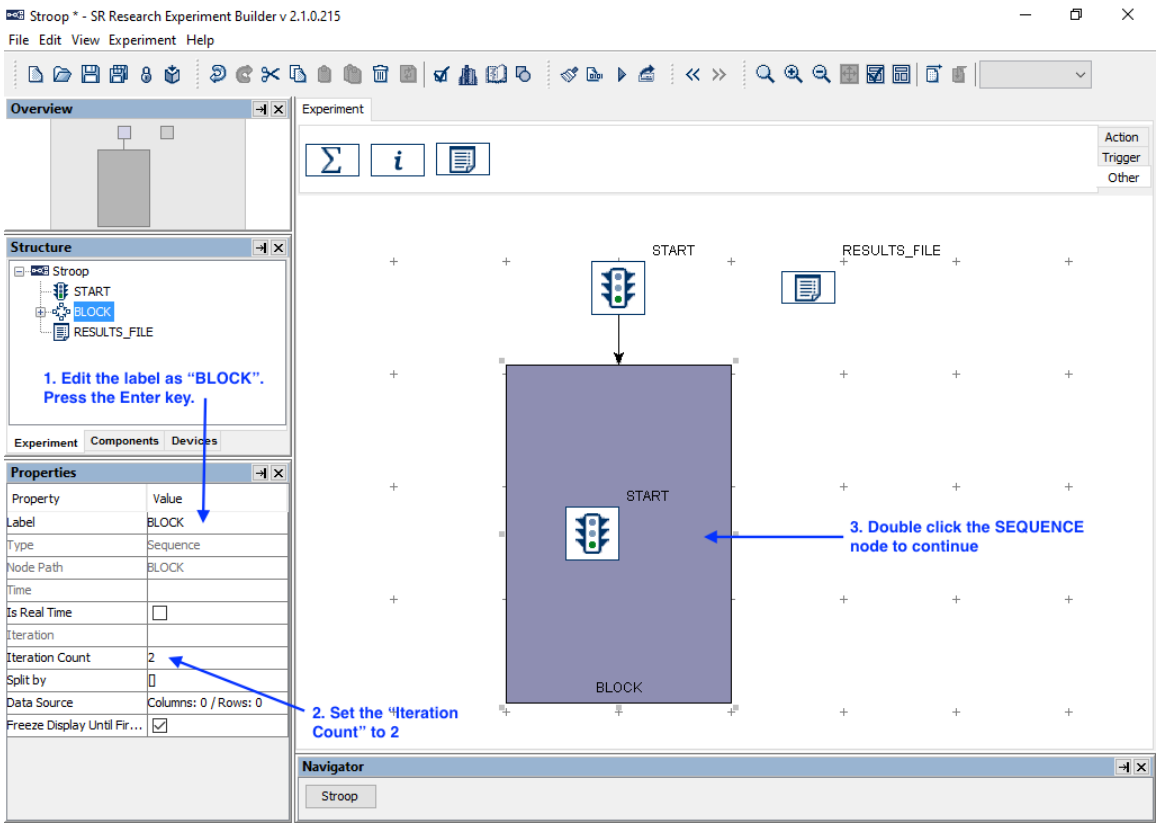

Figure 15-6. Editing Block Sequence.

1) Click the value field of the “Label” property of the Sequence created. Type a new label, e.g., “BLOCK”, into the text editor and press the Enter key to finish.
2) Click the “Iteration Count” value field and enter “2” as the total number of loops to execute.
3) Double-click the BLOCK Sequence object in the work space to enter the sequence.

In each block, we will first display some instructions onscreen and then run nine trials (see Figure 15-7). We can start by adding the necessary nodes to the workspace within the BLOCK sequence:

1) Open the “Action” Tab of the component toolbox, then click and drag a “Display Screen” action into the work area.
2) Open the “Trigger” Tab of the component toolbox, then drag a “Timer” trigger into the work space.
3) Select the Timer trigger and set the duration to 120000 msec.
4) Add a “Keyboard” trigger to the work space.
5) Open the “Action” Tab of the toolbox and add a “Sequence” node to the workspace. This will be our trial-level sequence.

Then we can continue by drawing the connections between the nodes in the BLOCK sequence:

6) Click and drag from the START node to the DISPLAY_SCREEN node.
7) Draw two connections from the DISPLAY_SCREEN action to the KEYBOARD and TIMER triggers. When a single action connects to multiple triggers, a number is added to each connection indicating the evaluation order among the two trigger types. In this experiment, it doesn’t matter which order the nodes are connected in.
8) Draw a connection from each of the two triggers to the trial-level SEQUENCE node.
9) Right-click any blank area in the work window and select "Arrange Layout" in the popup menu.

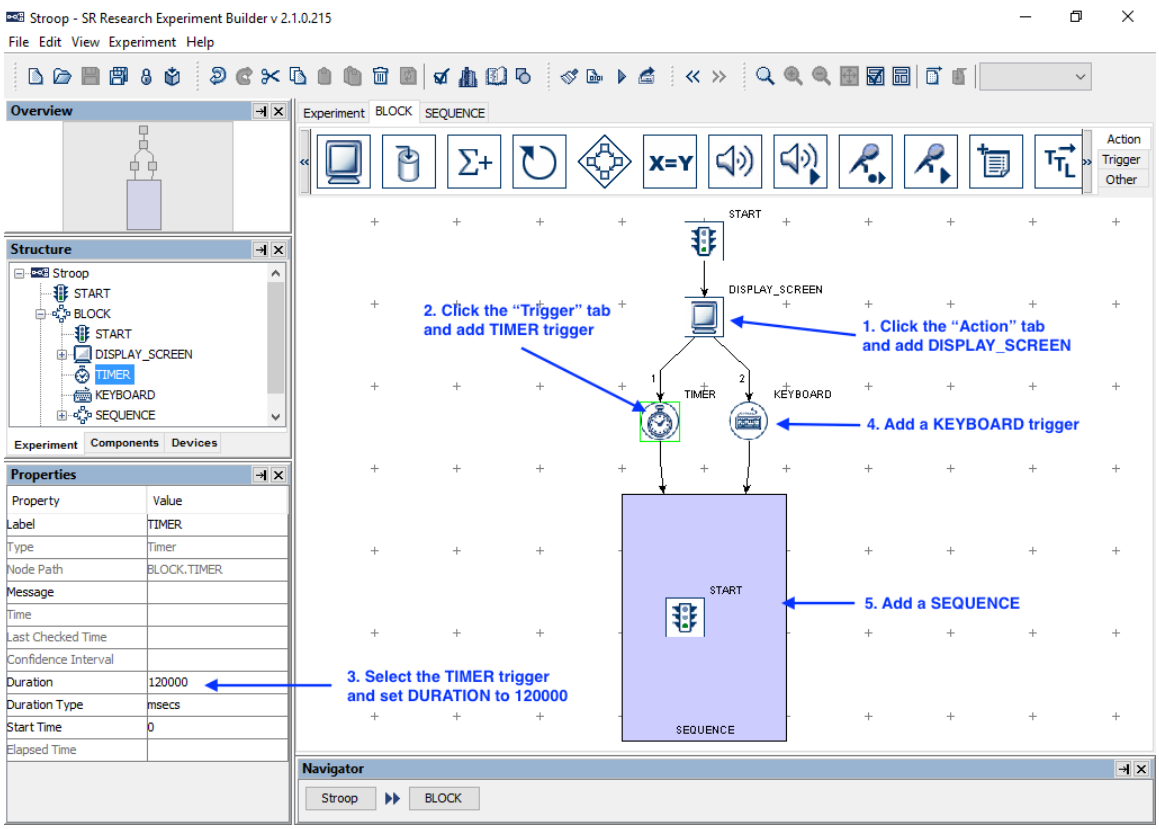

Figure 15-7. Adding Instruction to Block Sequence.

15.5 Creating the Instruction Screen

Next we can configure the DISPLAY_SCREEN node to display a set of instructions at the beginning of the experiment. In this example, we will create the instructions by adding a Multi-Line Text Resource to the Display Screen; users may also create the instructions as an image file and use the Display Screen to display the image.

To start editing the screen, open the Screen Builder by double-clicking the DISPLAY_SCREEN node in the workspace.

1) Once you've opened the Screen Builder, click the multiline text resource () button on the screen builder toolbar to select the resource to add.
2) Click anywhere on the screen to add the resource.

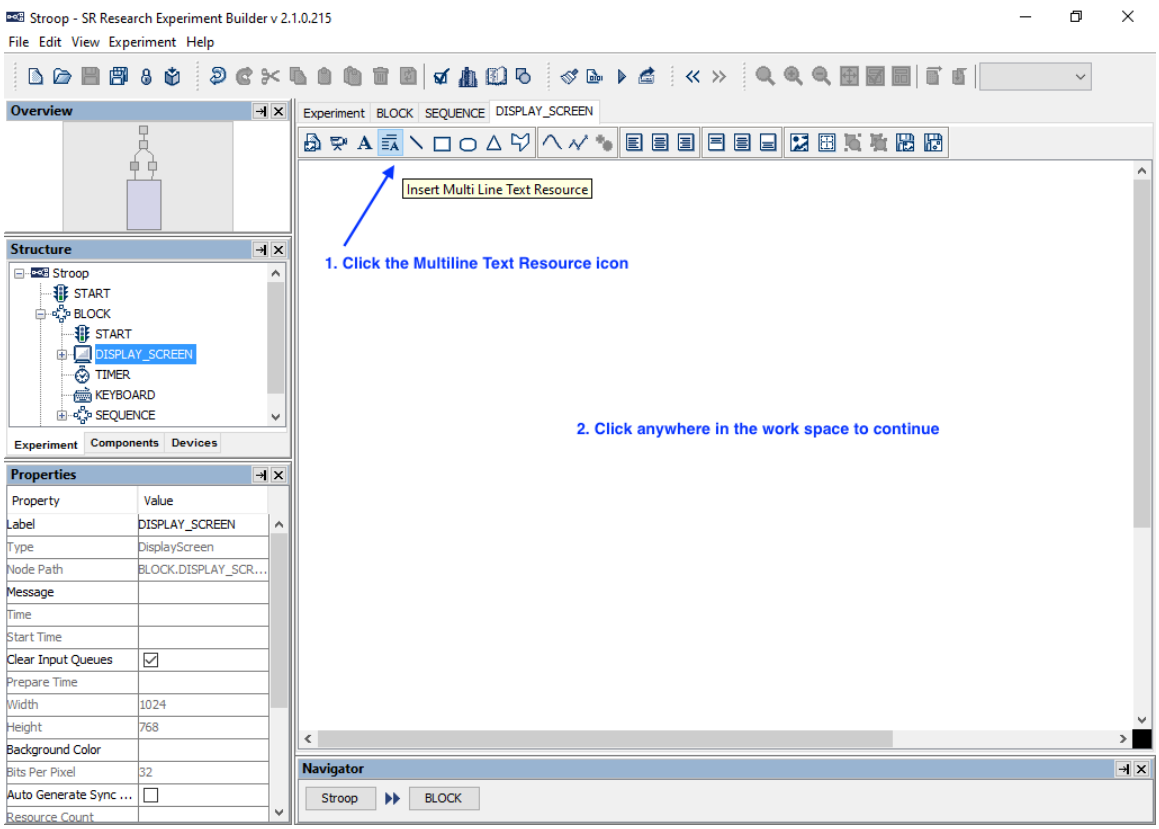

Figure 15-8. Adding Multiline Text Resource onto a Display Screen.

In the following Multiline Text Resource Editor,

1) Click the “Margins” button in the toolbar to set the text margins. Enter 100 in all fields. Click the "OK" button on the dialog box.
2) Enter the desired instruction text (see Figure 15-9).
3) Press Ctrl + A on Windows (Command ⌘ + A on macOS) to select all text entered.
4) Then click the buttons on the toolbar to set the desired text appearance (font name, font size, font style, alignment style, line spacing, and text color).
5) Select the example word “Green” in the text and set its color to blue and text size to 50.
6) Click the “Close” button ( ) at the top right corner to close the Screen Builder.

Please note that, instead of using a multi-line text resource, users can also present the instruction text by using an image resource (see Section 8.1.1).

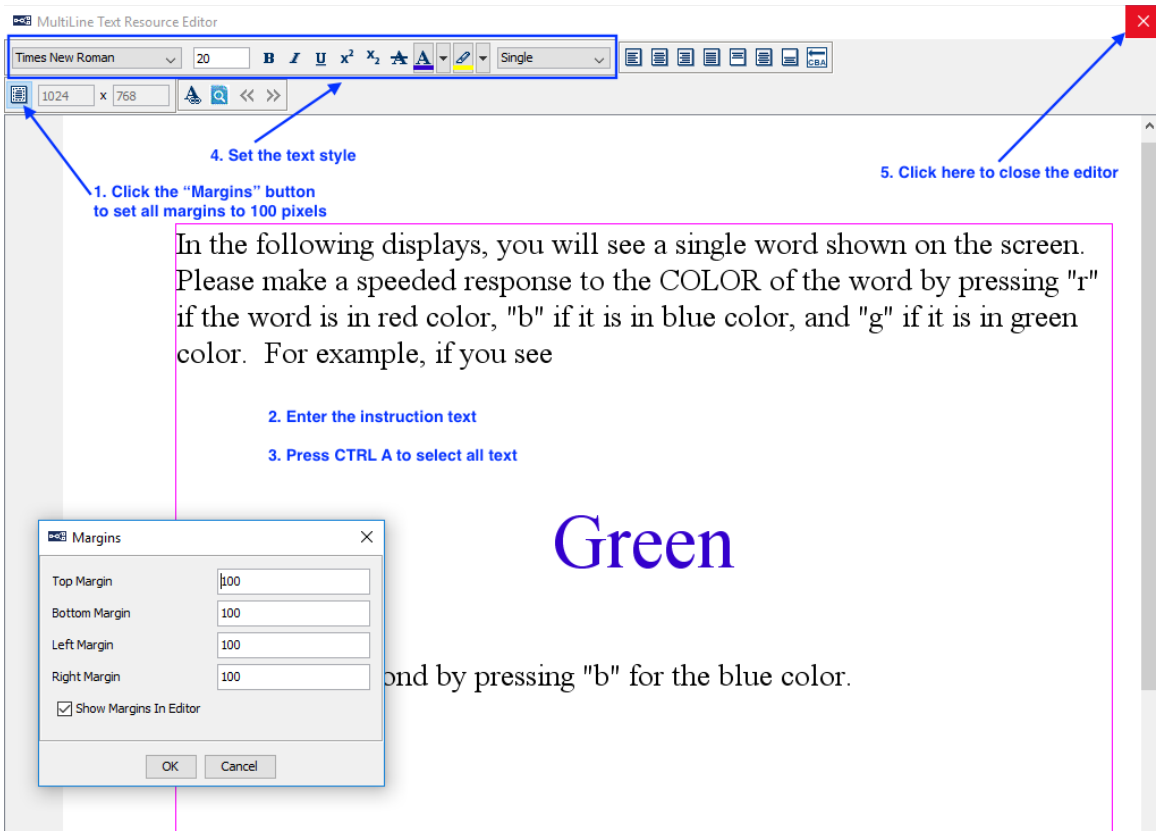

Figure 15-9. Create Instruction Screen.

15.6 Editing the Trial Sequence: Data Source

Next, we will design the Trial sequence, which will contain all the necessary triggers and actions in each trial. We will first create a data source to set the parameters in individual trials (see Figure 15-10). In this experiment, we will add four columns to the data source: "Color", to specify the color of the text; "Word", to specify the text displayed on each trial; "Expected", to specify the expected keyboard response; and "Compatible", to specify whether the text color matches the word.

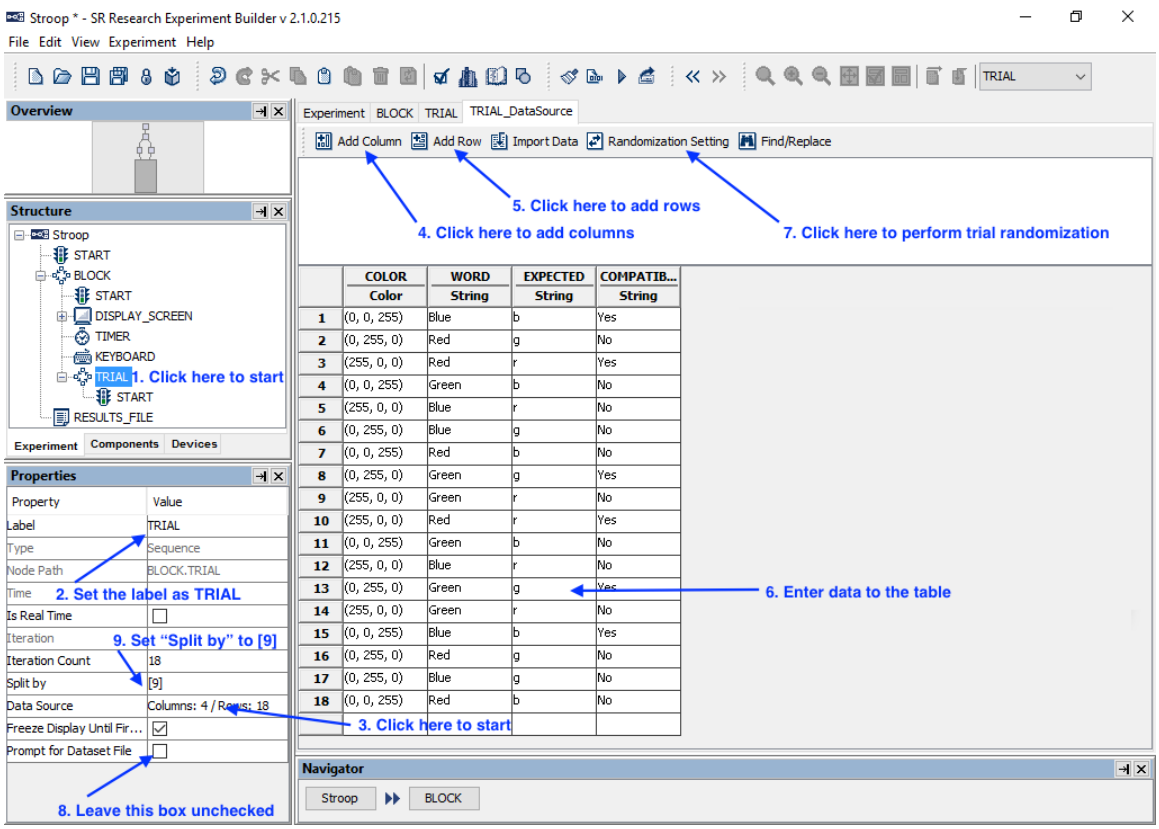

Figure 15-10. Creating Data Source.

1) First, select the last "SEQUENCE" node in the structure list (the one we added within the BLOCK sequence).
2) In the Properties table, enter a new value for the Label, e.g., "TRIAL".
3) Click the value cell of the "Data Source" property (where it says "Columns: 0 / Rows: 0") to bring up the Data Source Editor.

Next we can create the Data Source columns and enter our values.

4) Click the "Add Column" button. Type "COLOR" (without quotes) in the Column Name box, and set the Column Type to "Color". Click the "OK" button to finish. Click the "Add Column" button again, to create three more columns. Set the Column Names as "WORD", "EXPECTED", and "COMPATIBLE", and set these Column Types as "String". (**Important:** Your experiment may not run if inappropriate column types are used for the data source.)

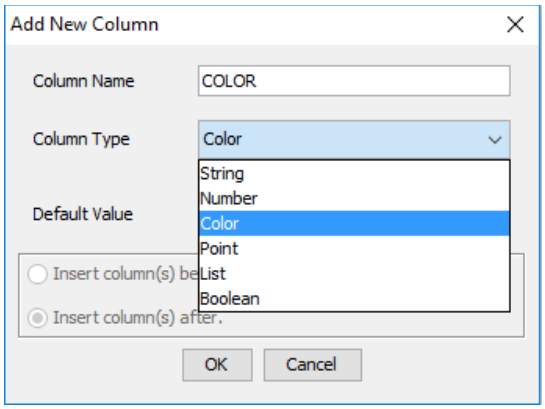

Figure 15-11. Adding a New Data Source Column.

5) Click the “Add Row” button. Set the “Number of Rows” to 18 to add 18 rows of empty cells, then click “OK”.
6) Add the following data to the table just created. For cells expecting Color data, enter the RGB values as three integers (between 0-255) separated by commas; Experiment Builder will automatically append a fourth alpha value of 255.

	Color	**WORD**	**EXPECTED**	**COMPATIBLE**
data type	Color	String	String	String
1	(0, 0, 255)	Blue	b	Yes
2	(0, 255, 0)	Red	g	No
3	(255, 0, 0)	Red	r	Yes
4	(0, 0, 255)	Green	b	No
5	(255, 0, 0)	Blue	r	No
6	(0, 255, 0)	Blue	g	No
7	(0, 0, 255)	Red	b	No
8	(0, 255, 0)	Green	g	Yes
9	(255, 0, 0)	Green	r	No
10	(255, 0, 0)	Red	r	Yes
11	(0, 0, 255)	Green	b	No
12	(255, 0, 0)	Blue	r	No
13	(0, 255, 0)	Green	g	Yes
14	(255, 0, 0)	Green	r	No
15	(0, 0, 255)	Blue	b	Yes
16	(0, 255, 0)	Red	g	No
17	(0, 255, 0)	Blue	g	No
18	(0, 0, 255)	Red	b	No

7) Click the "Randomization Setting" button to configure randomization settings. In the Randomization Setting window, set the randomization Seed Value to "Session Label" so the same trial sequence will be presented when the same recording session label is used. Check the "Enable Trial Randomization" box, and set the Run Length Control column to "WORD" and Maximum Run Length to 2. This ensures that the trial presentation order is randomized, with a restriction that the

same "WORD" value will not be shown on three consecutive trials. Press the OK button to finish.

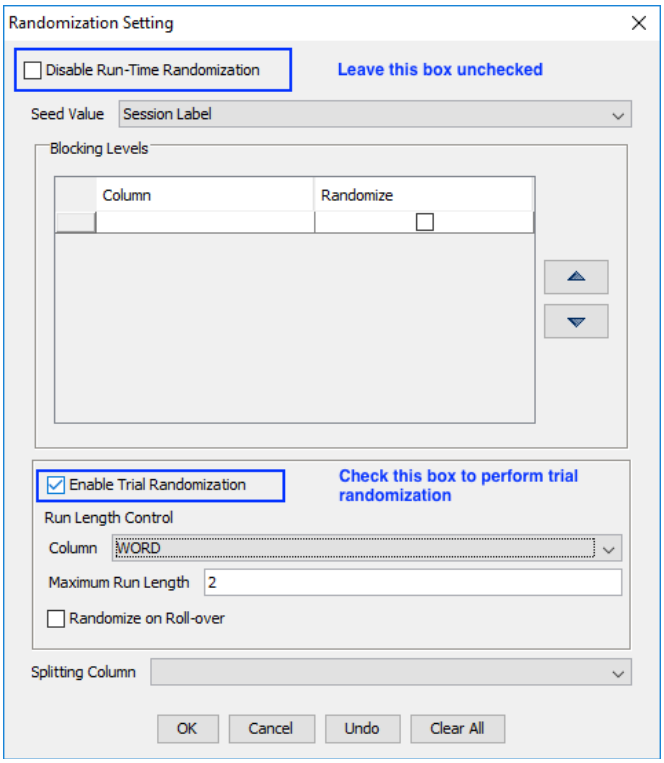

Figure 15-12. Data Source Randomization.

8) In the properties table of the TRIAL sequence, leave the "Prompt for Dataset File" box unchecked so you will not be prompted to select a data source file at the beginning of the experiment.
9) Click the "Split by" value field. Enter a value of [9] so 9 trials will be run in each block.
10) Double click the "TRIAL" sequence node in the workspace to enter the sequence.

15.7 Editing the Trial Sequence: Setting Initial Values and Preparing Sequence

Each trial should begin with a Prepare Sequence action, followed by the actual trial recording (see Figure 15-13). In a non-EyeLink experiment, the Prepare Sequence action preloads any image files or audio clips in the trial for real-time image drawing or sound playing and reinitializes trigger settings. We will also add several variables to store dependent data (e.g., RT, key press, and trial response accuracy) for each trial.

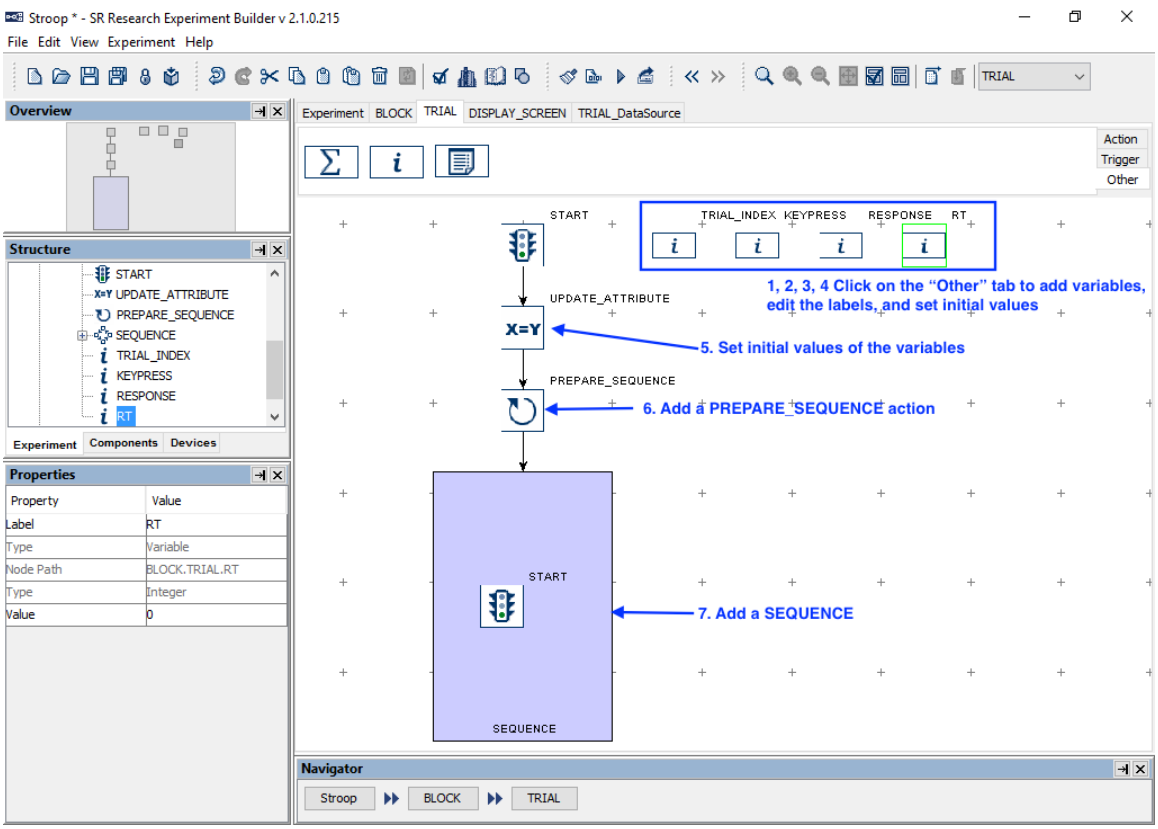

Figure 15-13. Editing Trial Sequence.

1) Open the “Other” Tab of the component toolbox and drag a “Variable” node into the work space. Select the newly added VARIABLE node, change its label to “TRIAL_INDEX” and set the initial value as “0”. This variable will be used to keep track of the current trial index. The data type will automatically changed to “Integer” upon setting the initial value of “0”.
2) Add another variable to the workspace, then set its label to “KEYPRESS” and its initial value to “.”. This variable will be used to record the participant’s key press for the trial.
3) Add a third variable, then set its label to “RESPONSE” and its initial value to “.”. This variable will be used to check whether the response recorded is correct or not.
4) Add in a fourth variable, then set its label to "RT" and its initial value to “0.0”. This variable will be used to store reaction time for the trial.
5) Open the "Action" Tab of the component toolbox, then drag an Update Attribute action into the work space. Select the Update Attribute node (step A1 of the figure below) and set the initial values of the following four variables, TRIAL_INDEX, RT, RESPONSE, and KEYPRESS:
    a. Click the value field of the "Attribute-Value List" property (step A2) to open the “Attribute-Value List" window.
    b. Click once in the first cell under the "Attribute" column (step A3), then click the […] button to open the Attribute Editor window. In the "Node Selection" panel on the left, select the TRIAL_INDEX variable node under the "TRIAL" sequence (step A4). In the middle "Node Attributes"

panel, double-click the "Value" attribute (step A5). This will set the contents of the "Attribute" panel on top as "@TRIAL_INDEX.value@". Click the "OK" button to finish. This will fill in the first cell of the Attribute-Value list dialog.

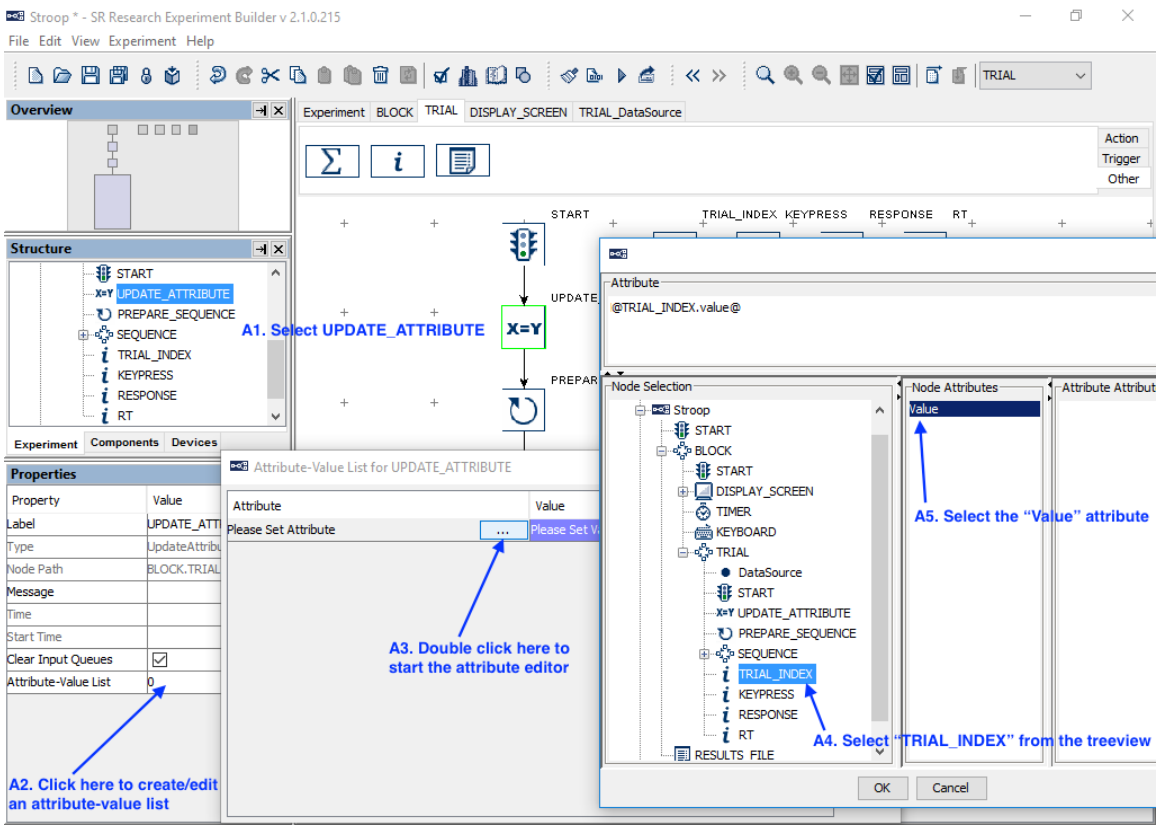

Figure 15-14. Updating Trial Index.

c. Next, click the first cell under the "Value" column once, then press the […] button (See B1 of the figure below). In the "Node Selection" panel of the attribute editor, click the "TRIAL" sequence (see B2). In the middle "Node Attributes" panel, double click the "Iteration" attribute (B3). This will update the contents of the top "Attribute" editor dialog as "@parent.iteration@". Click the "OK" button to finish (B4).

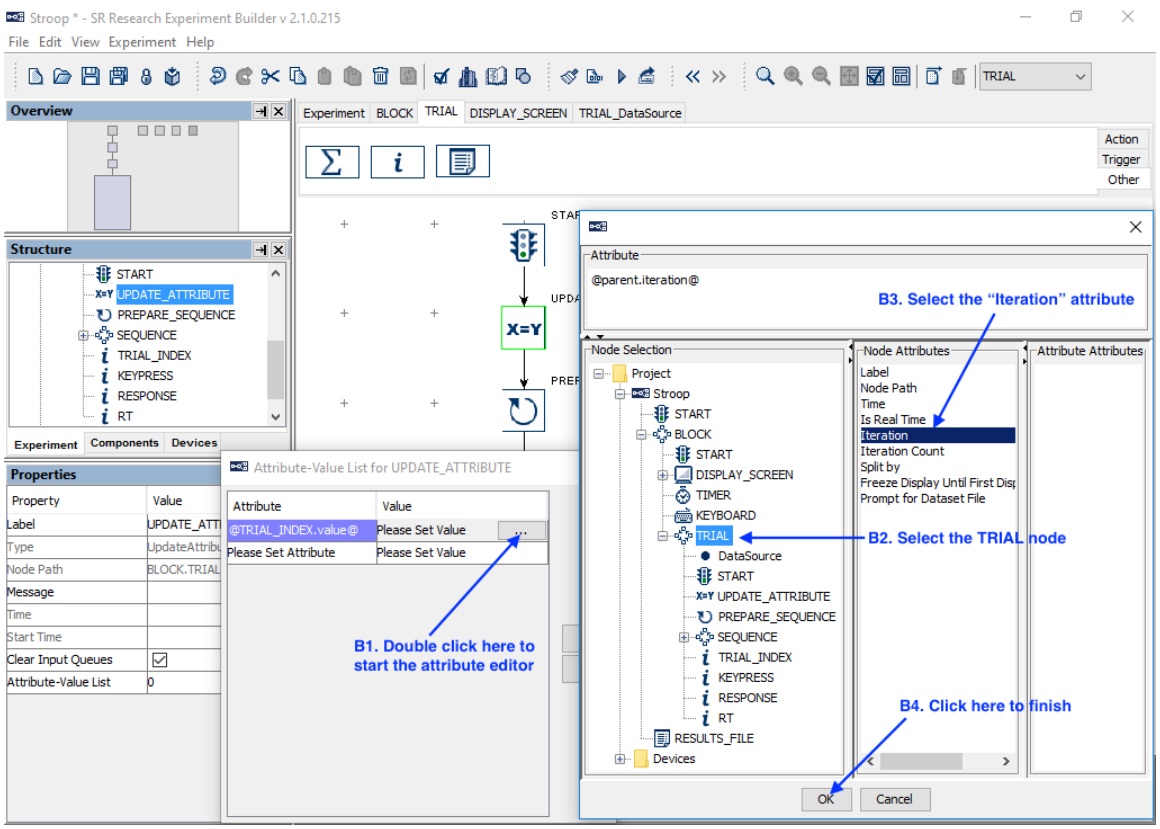

Figure 15-15. Update Trial Iteration.

d. Similarly, set the second cell of the "Attribute" column to @RT.value@. Double-click in the left side of the corresponding "Value" cell, then type in "-32768" (without quotes; this is equivalent to a "missing value" for Integer data) and press Enter to register the change (see Figure 15-16).
e. Set the Attribute 3 to "@KEYPRESS.value@" and value 3 to ".".
f. Set the Attribute 4 to "@RESPONSE.value@" and value 4 to ".".

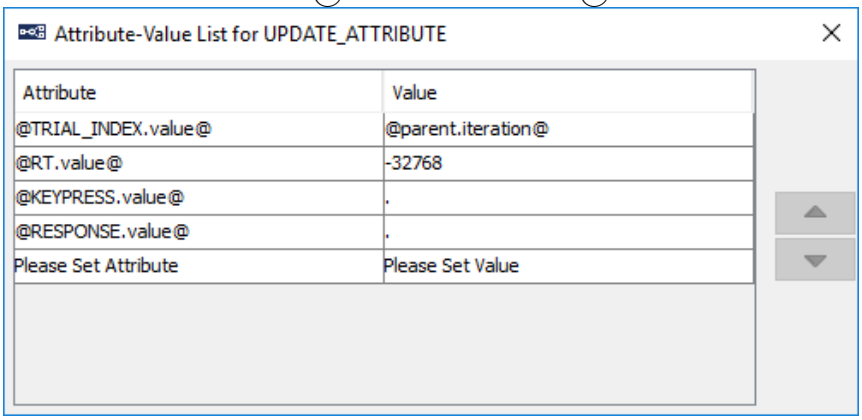

Figure 15-16. Updating the Attribute of RT.

6) Next, drag a “Prepare Sequence” action into the work space. Click the added action and review the settings in the property table—make sure the “Flush Log” box is checked so data output for the previous trial is completed before starting a new trial.
7) Then drag a “Sequence” node into the work space.
8) Click and drag to draw a connection from the “START” node to “UPDATE_ATTRIBUTE”, from “UPDATE_ATTRIBUTE” to “PREPARE_SEQUENCE”, and from “PREPARE_SEQUENCE” to the “SEQUENCE” node. The four variable nodes (RT, KEYPRESS, RESPONSE, and TRIAL_INDEX) should not be connected to other nodes.
9) Right-click any blank area in the workspace and select “Arrange Layout” in the popup menu.

15.8 Editing the Trial Event Sequence – Part 1

The next step is to design the actual display presentation in a trial. In this example, we first show a fixation cross in the center of the screen for one second, followed by the presentation of the Stroop word. We then wait for a keyboard response by the participant or for the trial to time out in eight seconds, and finally the display is cleared.

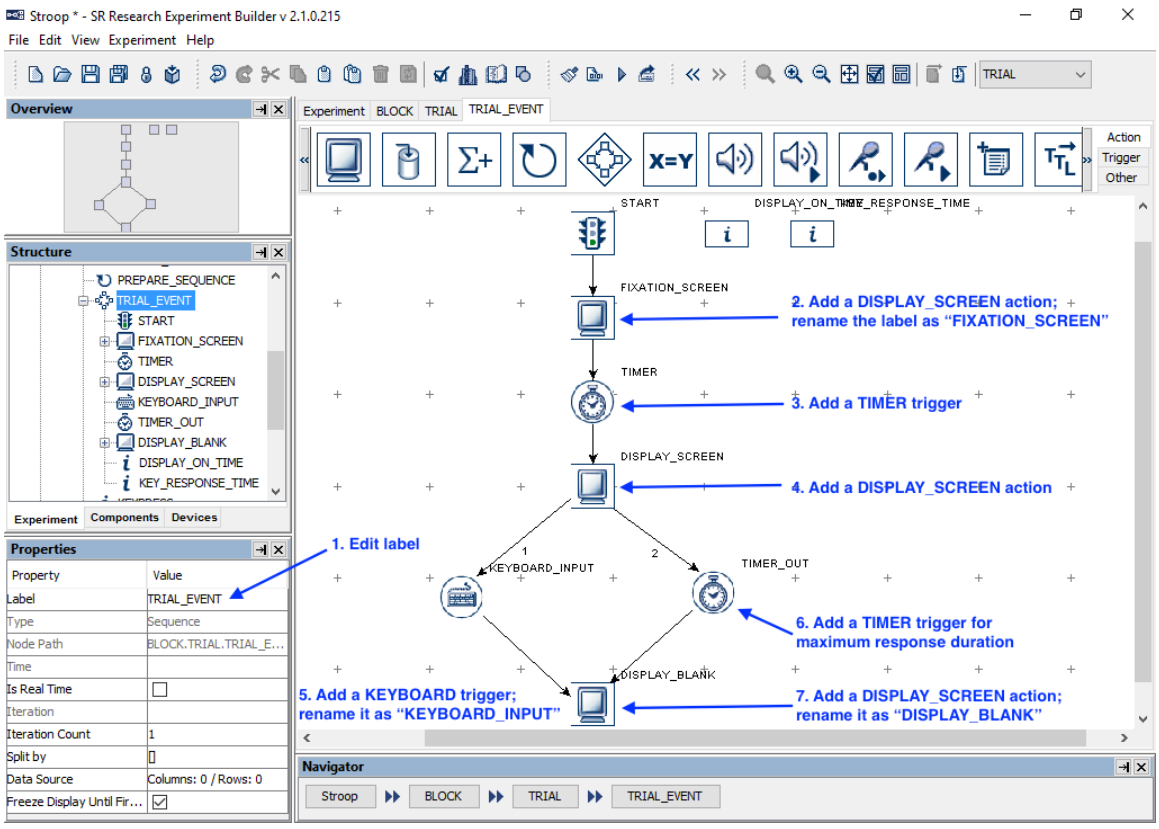

Figure 15-17. Editing Recording Sequence.

1) Select the newly added “Sequence” node and enter a new value for the Label, e.g., “TRIAL_EVENT”. If using an older computer with a single-core processor, make sure the “Is Real Time” box is **not checked,** as this may prevent the keyboard

from functioning. Then double-click the TRIAL_EVENT sequence in the workspace to enter the sequence.

2) Open the "Action" Tab of the component toolbox and drag a "Display Screen" action into the work area. Select the newly added Display Screen node and enter a new label, e.g., "FIXATION_SCREEN". Set the "Message" property as "fixation_display". We will later add a fixation cross in this screen (see Section 15.8.1).
3) Then open the "Triggers" tab and drag a "Timer" trigger into the work space. Select the Timer node, and enter "1000" in the "Duration" field.
4) Open the "Action" tab and drag a second "Display Screen" action into the workspace. This will be the screen showing the Stroop word (see Section 15.8.2). Set the "Message" property to "target_display" (without quotes) and press the Enter key.
6) Add a Keyboard Trigger. Select the trigger and set the Label, e.g., as "KEYBOARD_INPUT". Then set the "Message" property, e.g., as "keyboard_response". Double-click the left part of the value field for the "Keys" attribute to bring up a menu to choose the possible response keys. To select multiple response keys, hold the Ctrl key (Command ⌘ on macOS) and click on the desired keys. In this experiment, choose the following keys: b (for blue color), r (for red color), and g (for green color). Then click the "Close" button (☒) at the upper right corner of the dialog box to finish. The "Keys" property of the trigger will now be [b, g, r].

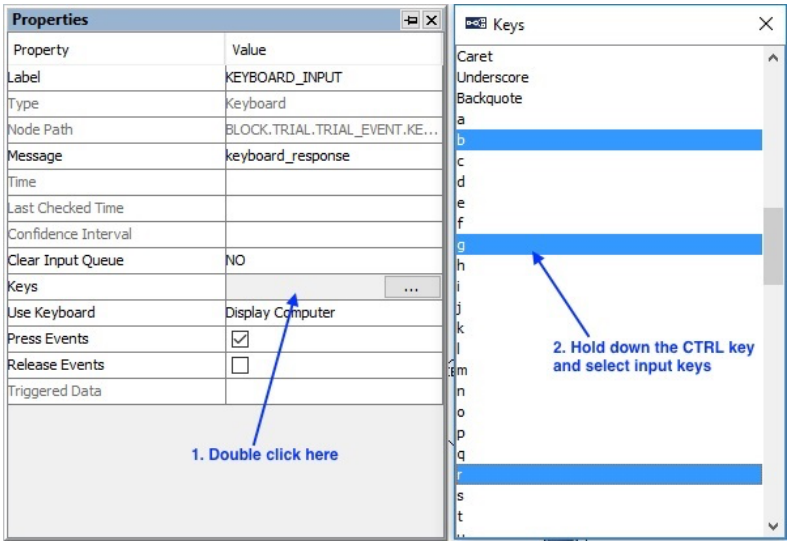

Figure 15-18. Setting Response Keys.

7) Add another Timer trigger to the work space. Set its Label as "TIME_OUT" and set the Duration as 8000 msec. Update the "Message" property of the trigger as "TIME_OUT".
8) Add another "DISPLAY_SCREEN" action and set its label as "DISPLAY_BLANK". This will be the blank screen to clear the Stroop word. Set the "Message" property as "DISPLAY_BLANK".

9) Draw a connection from the “START” node to “FIXATION_SCREEN”, from “FIXATION_SCREEN” to “TIMER”, from “TIMER” to “DISPLAY_SCREEN”, from “DISPLAY_SCREEN” to both “KEYBOARD_INPUT” and “TIME_OUT”, and from the last two triggers to “DISPLAY_BLANK”.
10) Right-click any blank area in the work space and select “Arrange Layout” in the popup menu to re-arrange the nodes.

15.8.1 Creating the Fixation Screen

We can now create the fixation screen. In this example, we will use an image to serve as the fixation target. (Alternatively, a text resource with a “+” may be used.) Images must be loaded into the Library Manager before they can be used. Follow the steps below to add the included image FIXATION.bmp to the resource library (see Figure 15-19):

1) From the application menu bar, select “Edit → Library Manager”.
2) In the Library Manager, select the “Image” tab.
3) Click the Add button to load in an image—the example image “FIXATION.bmp” can be found at “Documents\ExperimentBuilder Examples\Resources\Images\”. (Alternatively, click and drag the image file from Explorer in Windows or Finder in Mac into the Library Manager.) The image file properties and a preview of the image will then be displayed in the Library Manager.
4) Click the button on the Library Manager to finish.

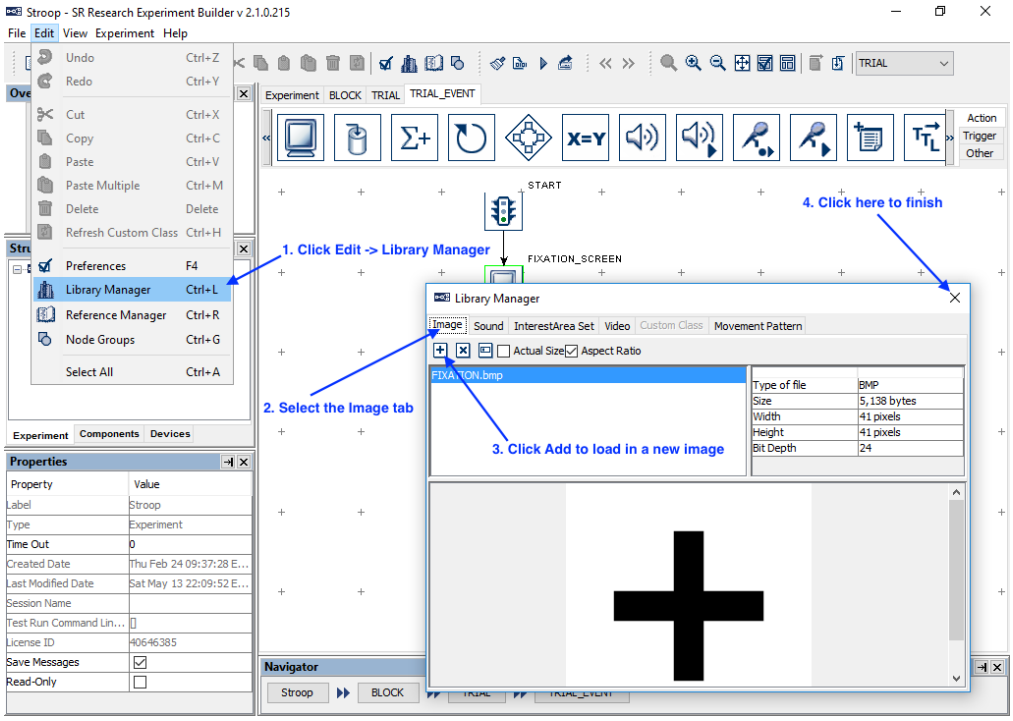

Figure 15-19. Loading Resources to Image Library.

5) Double click the FIXATION_SCREEN action in the graph workspace to open the Screen Builder.

6) Click the “Insert Image Resource” ( ) button on the Screen Builder toolbar, then click anywhere on the screen to add the image resource. In the following “Select Image” dialog, select “FIXATION.bmp”, then click the “OK” button.

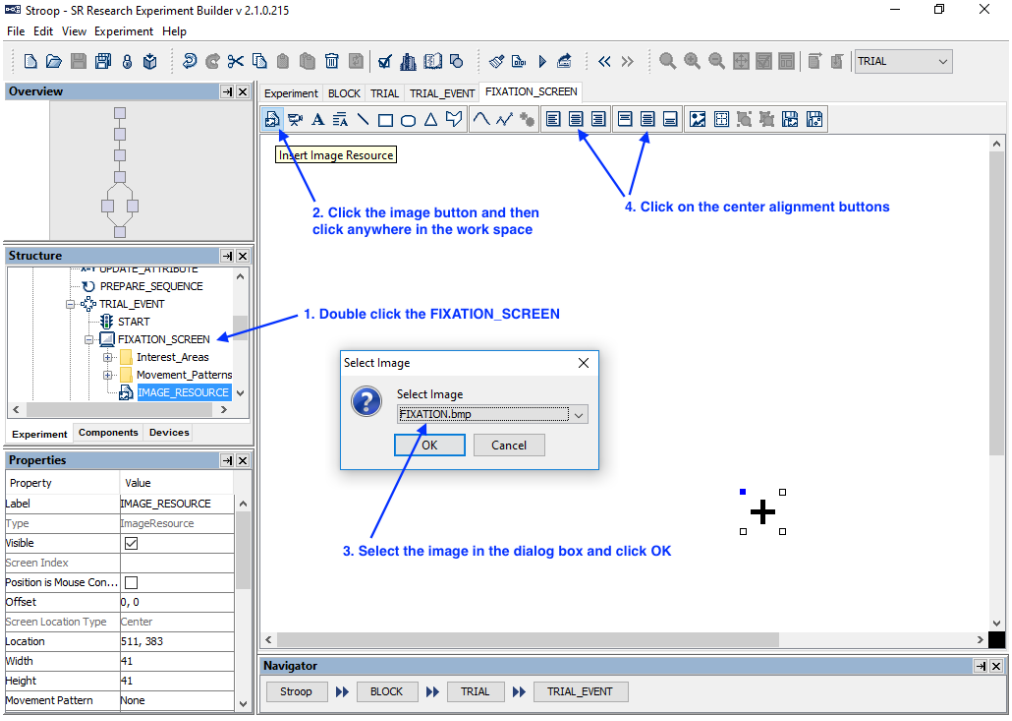

Figure 15-20. Loading Resources to Image Library.

7) Select the image resource. Click both the “Horizontal Center Alignment” and “Vertical Center Alignment” ( ) buttons to place it in the center of the screen. Right-click on the resource and choose “Lock Selection” in the popup menu to prevent the image from being accidentally moved in the Screen Builder.

15.8.2 Creating the Stroop Display Screen.

Next we will create a screen containing the Stroop color word. We will add a text resource to the display screen and modify the properties of the text resource, such as the text to be displayed, text color, font name, size, and alignment style. We will also create an interest area for the text (see Figure 15-21). To open the Screen Builder window, double-click the “DISPLAY_SCREEN” object in the work space.