Adding Brain Products Integration to Existing Non-EyeLink Experiments: A Quick Tutorial (EB Version 2.2.1)

This tutorial covers the steps required to add Brain Products EEG support to an existing experiment using version 2.2 or later of SR Research Experiment Builder. This example (BrainProducts_NonEyeLink_Stroop) is based on the "STROOP" example provided with Experiment Builder. Users who are new to the software are encouraged to re-create the STROOP example by following the step-by-step instructions provided in the Experiment Builder User Manual ("15 Creating Non-EyeLink Experiments: Stroop Effect"). This example illustrates using Experiment Builder to control the BrainVision Recorder without recording eye tracker data. If you are looking for an example that illustrates running experiments with simultaneous EEG and eye tracking recordings, please check out the "BrainProducts_EyeLink_Simple" example and the accompanying tutorial.

Experiment Builder implements integration with Brain Products EEGs using Remote Control Server 2.0 (RCS 2.0), an interface between TCP/IP and OLE automation interface commands in the BrainVision Recorder. This integration allows an EEG recording using the BrainVision Recorder to be controlled via the network. Make sure you have the Remote Control Server 2.0 (http://www.brainproducts.com/downloads.php) installed on the BrainVision Recorder computer. Experiment Builder automates the processes of opening and closing the EEG experiment sessions. Users can configure BrainVision Recorder Workspace and Experiment Number through the BRAIN_PRODUCTS device. With the BRAIN_PRODUCTS_CONTROL action, users can start, stop, pause, and resume EEG recordings, send event markers (through TTL) to mark stimuli and response onsets, set recorder modes (Monitoring, Impedance, View Test Signal), send annotations, and check the state of BrainVision Recorder, application, and acquisition.

Adding Brain Products RCS 2.0 support to an existing Experiment Builder project involves a few basic steps:

1) Configuring preferences in Experiment Builder,
2) Adding nodes to set the view mode of the BrainVision Recorder, and control the start and stop of the EEG recordings,
3) Checking the states (Application, Recorder, Acqusition) of the EEG recording
4) Sending event markers (through TTL) to the BrainVision Recorder, and
5) Configuring network settings for the Display PC that runs Experiment Builder and the BrainVision Recorder PC.

This tutorial assumes that the user already has the basic experiment programmed and tested. Although the discussion is based on the Stroop example, steps covered here can be easily applied to any non-eyetracking experiments programmed with Experiment Builder.

Please report all functionality comments and bugs to support@sr-research.com.

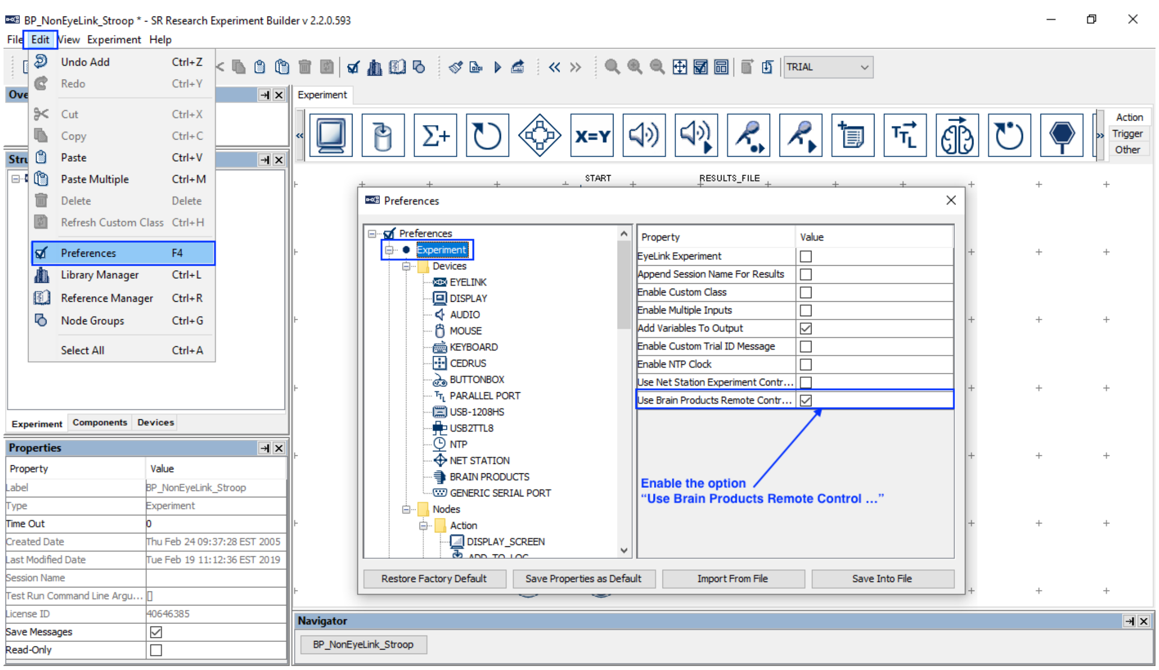
1 Configuring Experiment Builder Preferences

The integration between Experiment Builder and Brain Products EEGs uses Remote Control Server 2.0 (RCS 2.0), an interface between TCP/IP and OLE automation interface commands in the BrainVision Recorder. Users first need to configure the following Experiment Builder preferences to enable the components related to the Brain Products integration and to establish proper connections between computers.

1.1 Enabling Brain Products Remote Control Interface

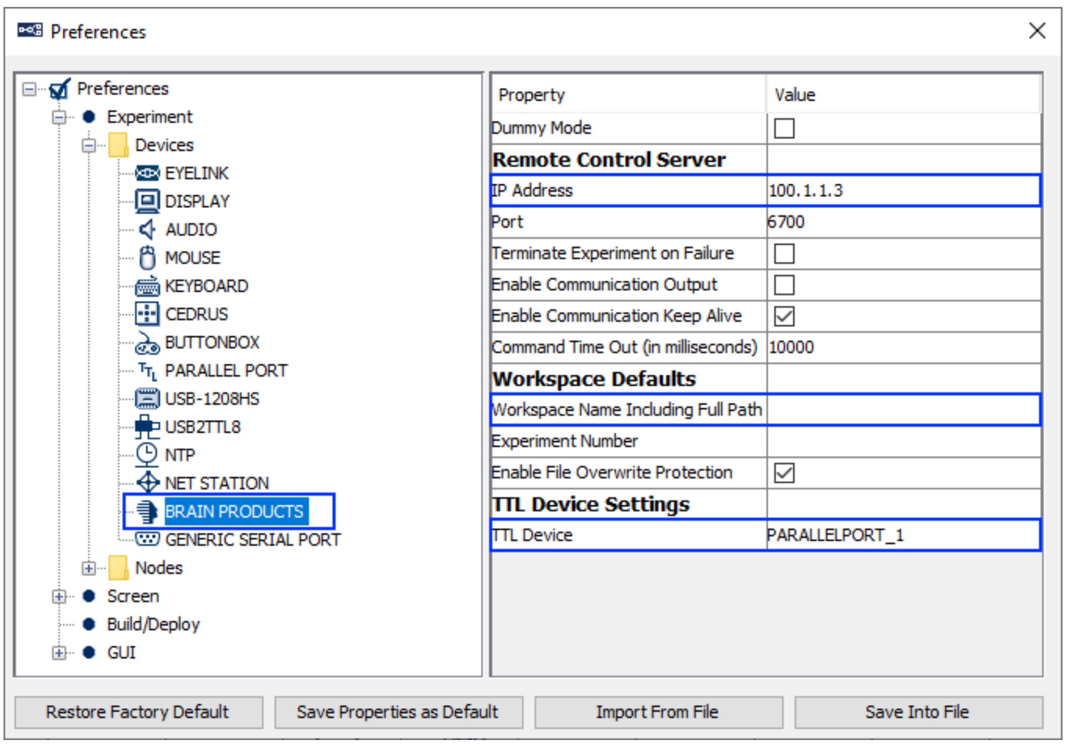
Follow the steps below to enable the Brain Products Remote Control Interface.

1) Select “Edit -> Preferences” from the application menu bar or press the shortcut key “F4” on Windows.
2) Click “Preferences -> Experiment” and check the “Use Brain Products Remote Control Interface (RCS 2.0)” option. Toggling this option will enable the “Brain Products” device and the “BRAIN_PRODUCTS_CONTROL” action for the project.



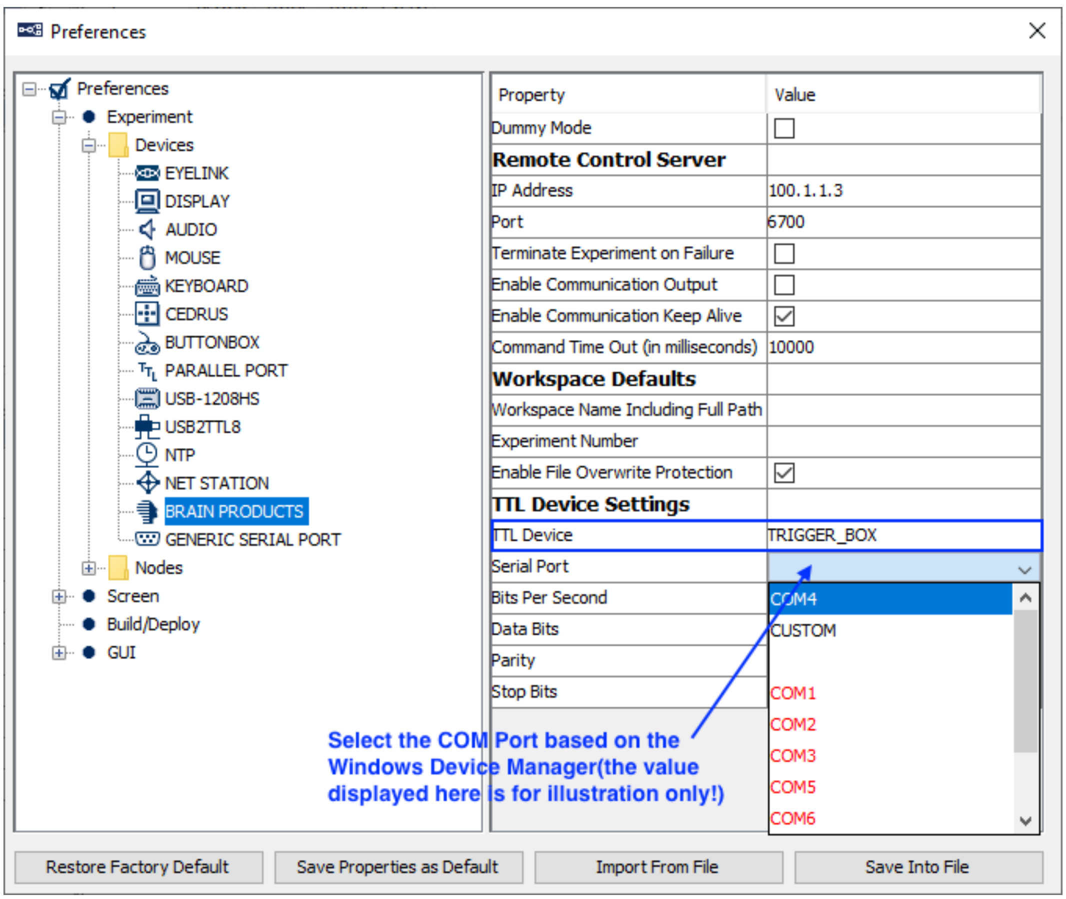
1.2 Review the Brain Products Device Settings

To configure the Brain Products device, go to "Preferences -> Experiment -> Devices -> Brain Products". Users need to fill in the proper IP address of the computer that BrainVision Recorder computer to establish the TCP/IP connection between the display computer and BrainVision Recorder computer. In this example, the IP address of the BrainVison Recorder computer was set to 100.1.1.3 with the subnet mask being 255.255.255.0, assuming the display (Experiment Builder) computer uses the standard IP address of 100.1.1.2. Please see section 5.1 for instructions on configuring the IP address of the Experiment Builder computer and/or BrainVision Recorder computer. The "Dummy Mode" option may be enabled to test run projects without an actual connection to the BrainVision Recorder computer. Users should make sure this option is turned off before running the actual EEG experiments.



In this example, the "Path to Workspace" is left blank. If no workspace is supplied, Experiment Builder will use the last workspace used in the BrainVision Recorder. When running the experiment, after entering the session name you will see a "Confirm Brain Products Workspace path" dialog box that displays the workspace to be used. While users may enter the desired workspace path manually at runtime, it is best to properly configure the Workspace Name Including Full Path property to avoid user error.

In this example, the TTL event marker signals are sent through the parallel port on the Display PC using a BRAIN_PRODUCTS_CONTROL action. Users can set the "Parallel Port One Base Address" of the Parallel Port device to 0x0 so that Experiment Builder automatically detects the base address of the parallel port on the Display PC. For display computers without a parallel port, users have other options to send TTL signals. For example, the USB-1208 HS by Measurement Computing and the USB2TTL8 by LabHackers are USB-based TTL devices that can be used on both a Mac and Windows PC. In an EyeLink experiment, users may send TTL signals through the devices installed on the EyeLink Host PC (configured through the "EyeLink_Host_TTL" Device). Experiment Builder also supports the TriggerBox from Brain Products when a Windows PC is used (https://www.brainproducts.com/productdetails.php?id=55). Please make sure the driver of the TriggerBox is installed. The COM port that the TriggerBox uses can be found in the "Ports (COM & LPT)" section of the device manager. Search the entry "TriggerBoxVirtualSerial Port (COMx)". In the BRAIN_PRODUCTS device, set the "TTL device" to "TriggerBox" and then select the appropriate COM port from the drop-down list.



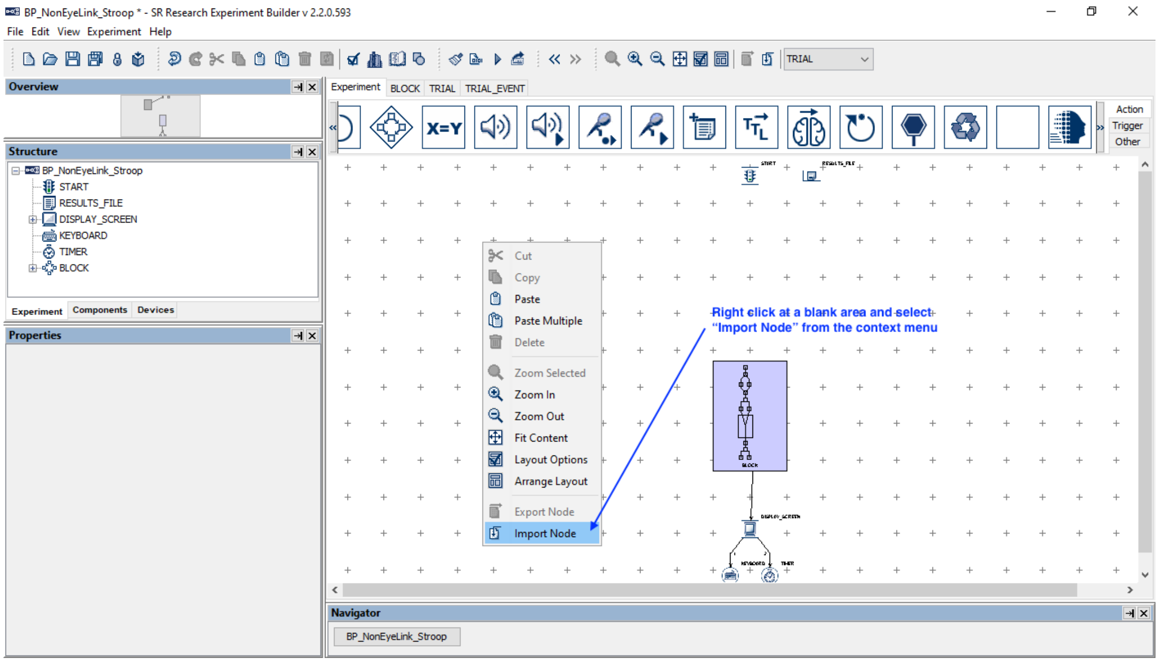
2 Setting BrainVision Recorder Mode and Controlling Recordings

Experiment Builder does the Brain Products integration by automatically connecting to the Remote Control Server 2.0 (RCS 2.0), opening the BrainVision Recorder application, initializing the workspace and setting the experiment name and subject ID at the beginning of the experiment. At the end of the experiment, Experiment Builder sends a command to close the BrainVision Recorder and then disconnects. Users can add BRAIN_PRODUCT_CONTROL actions into the project to control other critical aspects of the implementation, such as getting EEG ready for recording and setting the mode of the BrainVision Recorder (section 2.1); starting and stopping EEG recordings, and sending synchronizarion signals at the beginning and end of the experiment (section 2.2); checking the EEG recording states (section 3); and sending event markers to the Recorder (section 4).

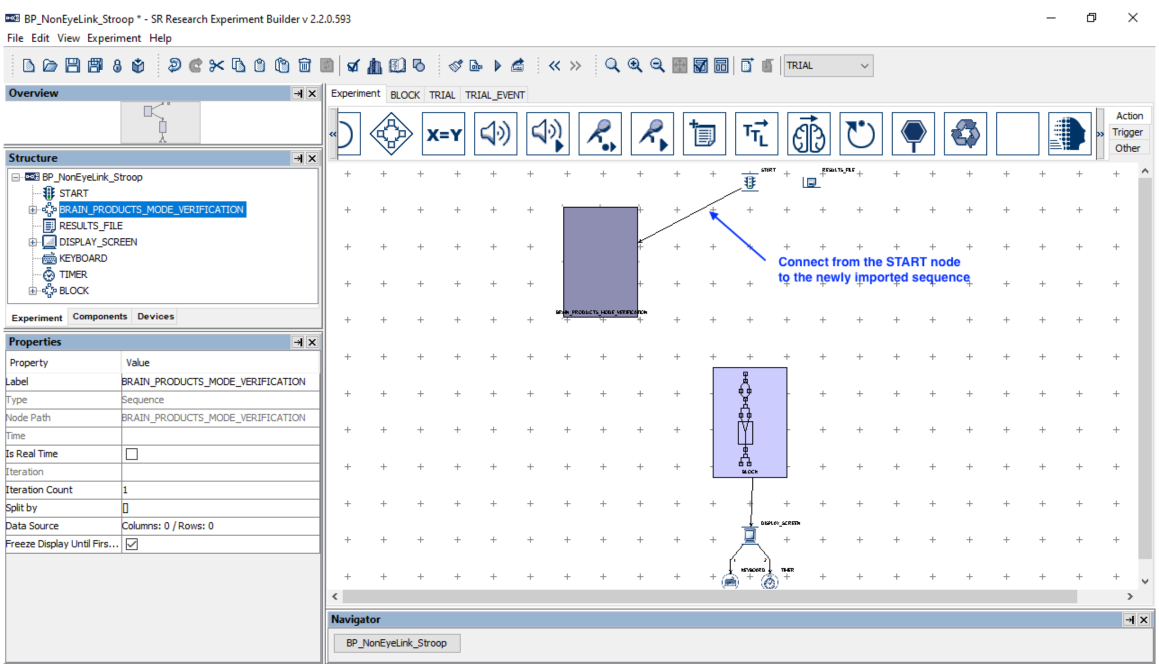
2.1 Getting EEG Ready for Recording and Setting BrainVision Recorder to Monitoring Mode

Experiment Builder connects to the RCS 2.0 and turns on the BrainVision Recorder when the user opens the .exe file to run an experiment. For some setups, however, the EEG system may not be ready for recording immediately when the experiment starts. (For example, if using LiveAmp, while the user can start Recorder over RCS, they will still need to manually search for and connect to LiveAmp.) The EEG must be ready before it can be set into the “Monitoring” mode to record data. Therefore, the first step in an experiment will be displaying some instructions to the experimenter to get the system ready. Since most of the steps in getting the EEG ready and checking the mode initialization will be the same across all experiments (but can be customized for each experiment), Experiment Builder simplifies these by including a re-usable module in the “EEG Integration” folder that can be shared across experiments.

1) Go to the topmost layer of the experiment. Click anywhere in the blank area of the workspace to make sure that no node or sequence is selected. Click the right mouse button and select "Import Node". (If the "Import Node" option is grayed out, please make sure no node is currently selected.) In the following "Open" dialog box, go to “ExperimentBuilder Examples\EEG Integration”, select the “BRAIN_PRODUCTS_MODE_VERIFICATION.ebo” file, and click "Open".



2) Make a connection from the START node to the newly imported BRAIN_PRODUCTS_MODE_VERIFICATION sequence.

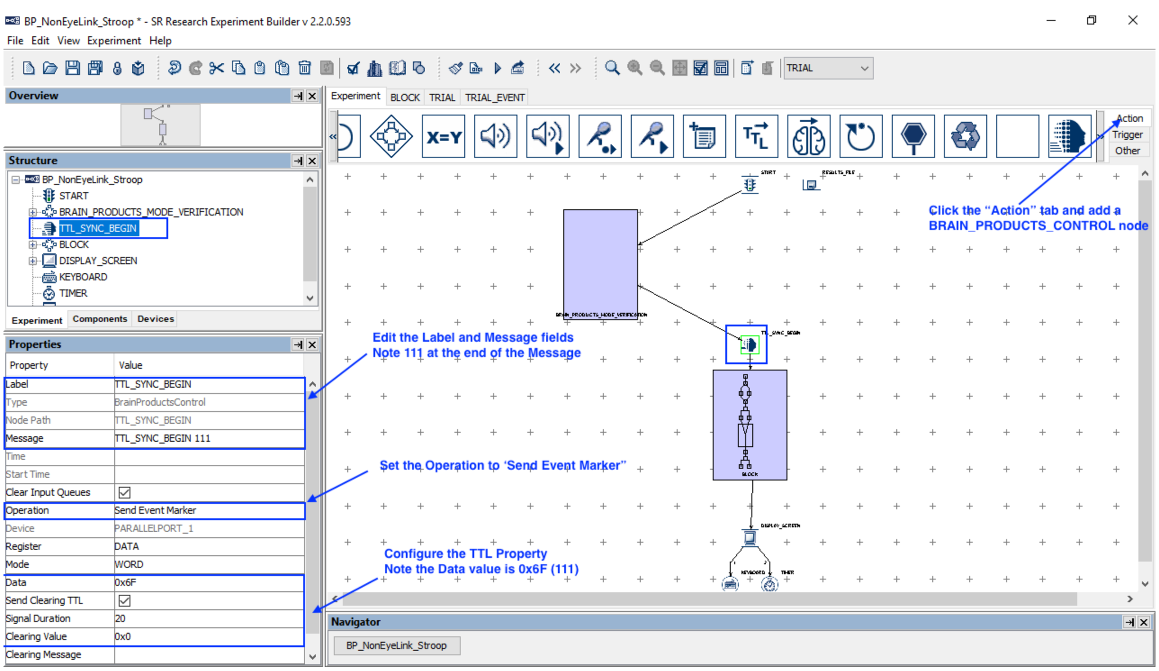


Users can double-click the BRAIN_PRODUCTS_MODE_VERIFICATION sequence to view the contents. This sequence contains the mode verification steps typically used used when setting up the EEG. To record EEG data, the BrainVision Recorder must be in the "View Data (Monitoring)" mode. The "Set Mode" operation of the BRAIN_PRODUCTS_CONTROL action allows the user to set the BrainVision Recorder to the "View Data (Monitoring)" mode at the beginning of the experiment. Since it may take a few seconds for the Recorder to switch to the proper mode, it is best to present instructions using a DISPLAY_SCREEN action right before the first BRAIN_PRODUCTS_CONTROL node so the experimenter is aware of the mode change. The BRAIN_PRODUCTS_CHECK_STATE nodes and CONDITIONAL trigger are used to check the current state of the EEG system so that users can either proceed with experiment recording or go back to initial screen to trouble shoot setup issues.

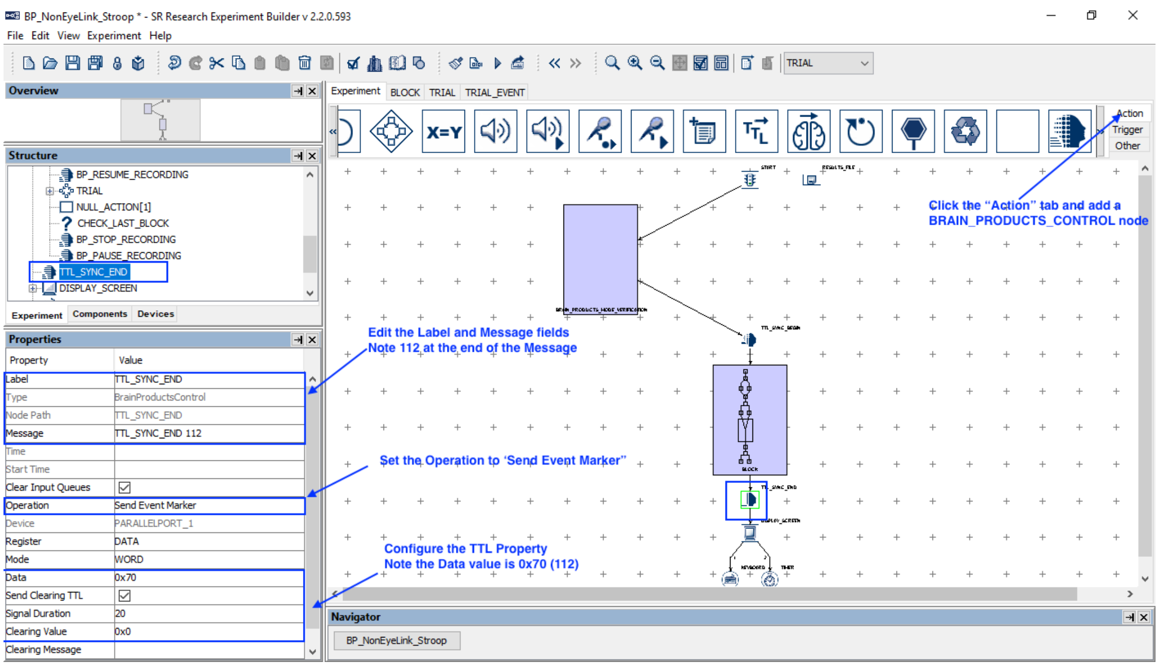
2.2 *Sending Synchronization Signals*

To align the EEG data with the events recorded in Experiment Builder, we will send a pair of synchronization pulses to the EEG recorder, one at the beginning of the experiment, and one at the end.

1) Select the "Action" tab of the Component Toolbox and add a BRAIN_PRODUCTS_CONTROL action. Set the Label of the action to "TTL_SYNC_BEGIN" and the message of the action to "TTL_SYNC_BEGIN 111"—the "111" corresponds to the TTL value that will be sent. Set the "Operation" to "Send Event Marker". Set the "Data" to 0x6F (111 in decimal). The "Send Clearing TTL" box is checked, the Signal Duration is set to 20 (ms) and a clearing value of 0x0 is used.. Draw a connection from the BRAIN_PRODUCTS_MODE_VERIFICATION sequence to the TTL_SYNC_BEGIN node and from TTL_SYNC_BEGIN to the BLOCK sequence.



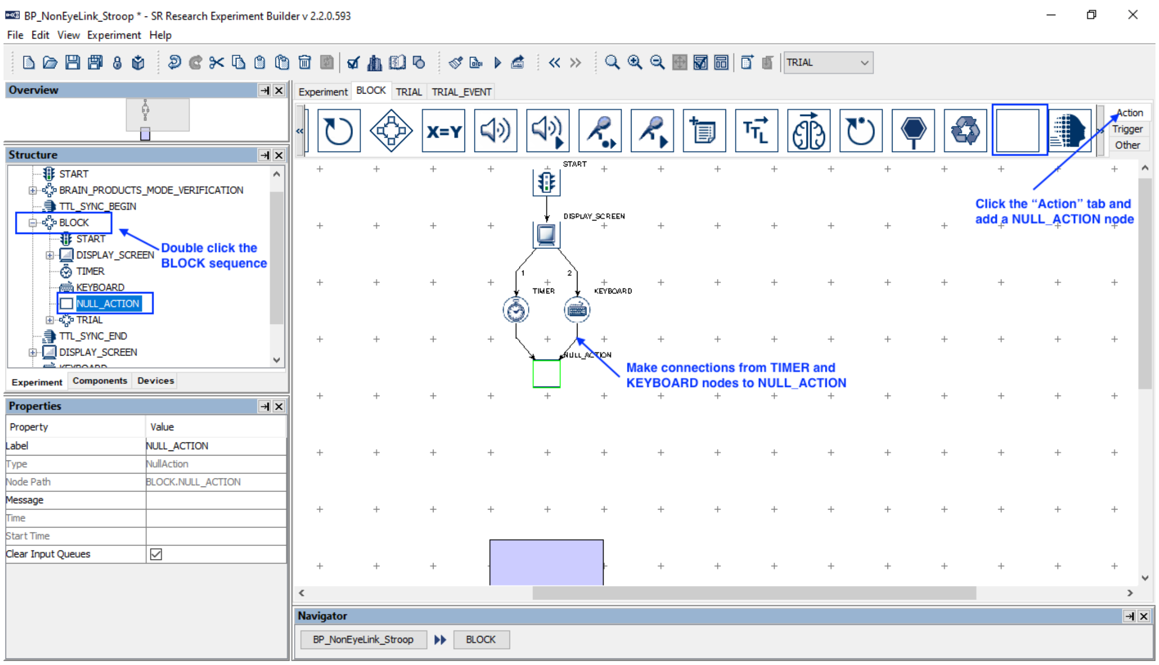
2) A sync signal will also be sent at the end of the experiment. Select the “Action” tab of the Component Toolbox and add a BRAIN_PRODUCTS_CONTROL action. Set the Label of the action to “TTL_SYNC_END” and the message of the action to “TTL_SYNC_END 112”—as before, the “112” corresponds to the TTL value that will be sent from this action. Set the “Data” to 0x70 (112). The “Send Clearing TTL” box is checked, the Signal Duration is set to 20 (ms) and a clearing value of 0x0 is used. Draw a connection from the BLOCK sequence to the TTL_SYNC_END node and from TTL_SYNC_END to the DISPLAY_SCREEN action.



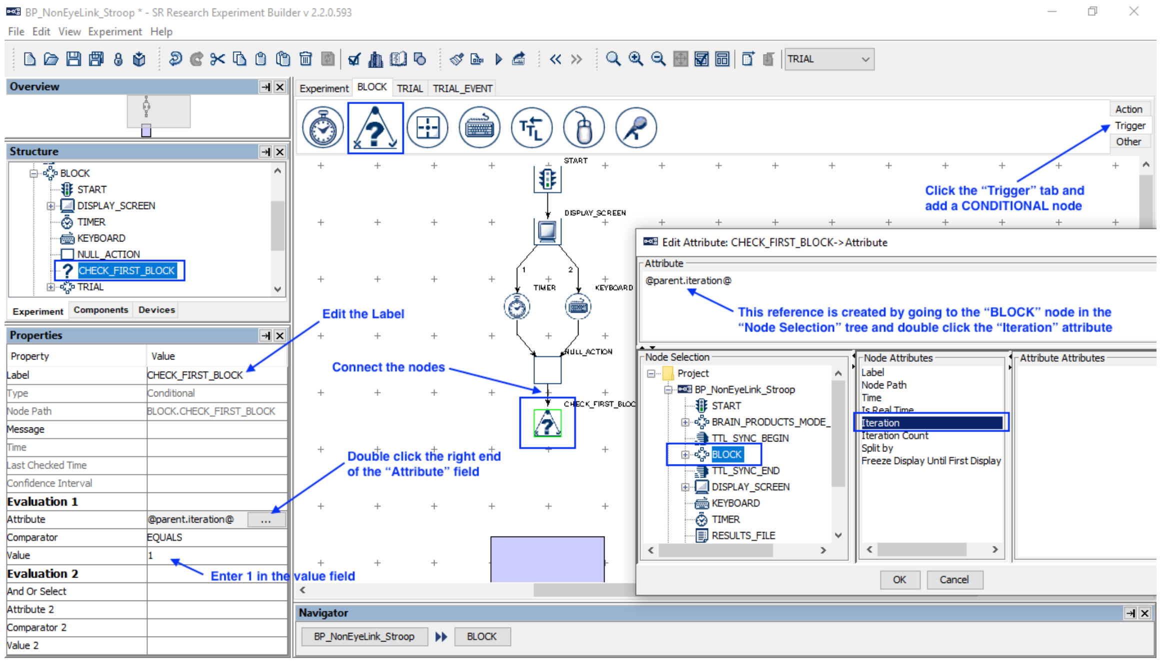
2.3 Starting and Stopping EEG Recordings

The “START RECORDING” and “STOP RECORDING” options of the Recording Control operation of the BRAIN_PRODUCTS_CONTROL action allow users to control EEG recordings. In a typical EEG session, users will need to start and stop recording only once, using a “START RECORDING” action at the beginning of the experiment and a “STOP RECORDING” action at the end of the experiment. Users can pause the recording following a block of trials, and then unpause it prior to the next block of trials. It is not recommended to start and stop EEG recordings too often (e.g., at the trial level). This example implements the EEG recording control by starting the recording at the beginning of the first block, pausing and resuming recordings between blocks, and then stopping recording at the end of the last block. This will create individual EEG recording files on the BrainVision Recorder for each of the blocks. (A simpler implementation would be to include a BRAIN_PRODUCTS_CONTROL action to start recording before the BLOCK sequence and another BRAIN_PRODUCTS_CONTROL action at the end of the experiment to stop recording, but this may record an unnecessary large amount of data if there are long breaks between blocks. See the Brain_Products_Simple example for an illustration of experiment-level recording.)

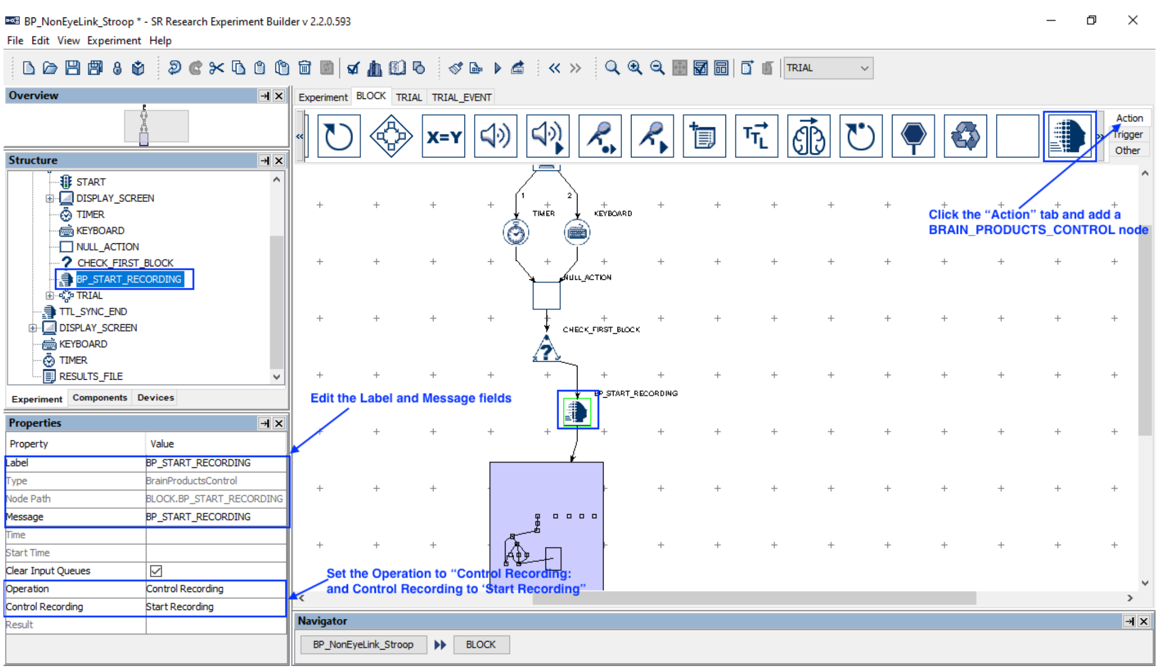
1) Double click the BLOCK sequence.
2) Select the “Action” tab of the Component Toolbox, click the NULL action ( ), and drag the node into the graph. This action doesn’t do anything except for making it possible to add a trigger following the existing TIMER and KEYBOARD triggers. Make a connection from both the TIMER and KEYBOARD nodes to the NULL_ACTION node.



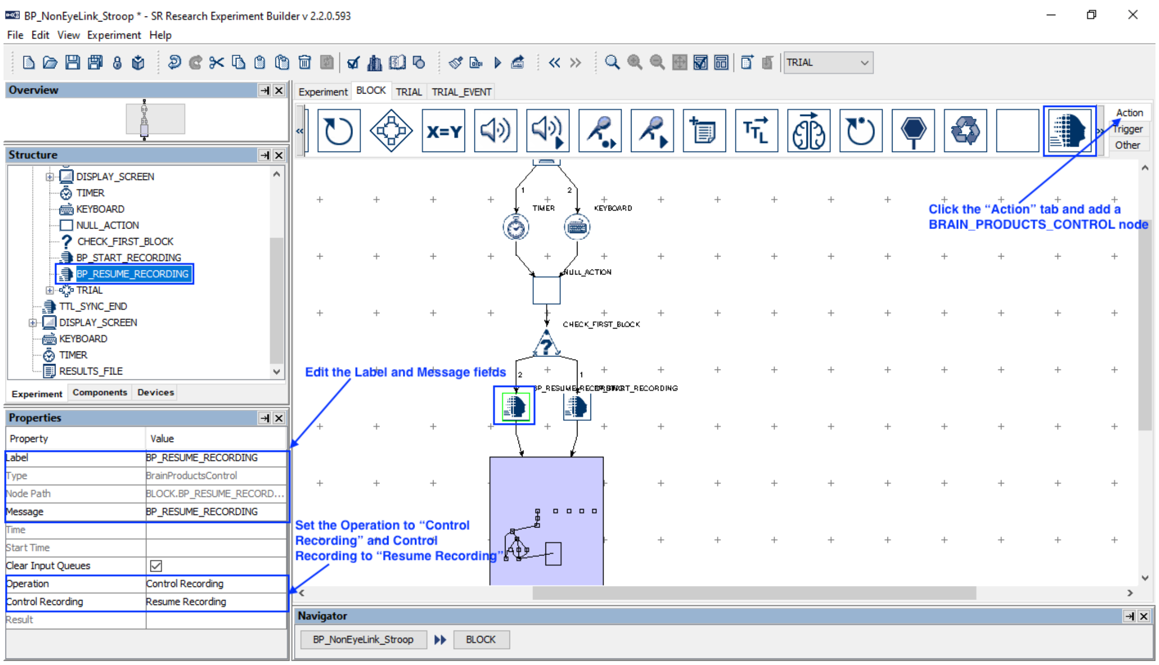
3) First we will add a CONDITIONAL trigger to check the block iteration. Select the “Trigger” tab of the Component Toolbox, click the CONDITIONAL trigger ( ), and drag the node into the graph. Make a connection from the NULL_ACTION node to the CONDITIONAL node. Select the newly added CONDITIONAL trigger. Rename the “Label” as “CHECK_FIRST_BLOCK”. Double click the right end of the “Attribute” field. This brings up an attribute editor dialog box. In the “Node Selection” panel on the left, find the BLOCK node and double click the “Iteration” node in the “Node Attributes” panel in the middle. This will create a reference @parent.iteration@. Please don’t manually type in the reference yourself! Click the OK button to close the dialog box. Set the “Comparator” to “EQUALS” and “Value” to 1. This conditional evaluation checks whether the current block iteration is 1. If true, we will start the EEG recording when the block starts and pause the recording at the end of the block. If not (i.e., for the subsequent blocks), we will resume the recording at the beginning of the trial block.



4) Select the "Action" tab of the Component Toolbox and add a BRAIN_PRODUCTS_CONTROL node to the graph. Edit the "Label" and "Message" properties of the node to "BP_START_RECORDING". Set the "Operation" to "Control Recording", and then choose "Start Recording" option from the "Control Recording" drop-down list. Now draw a connection from the right branch of the CHECK_FIRST_BLOCK conditional trigger to BP_START_RECORDING, and from the BP_START_RECORDING node to the TRIAL sequence.



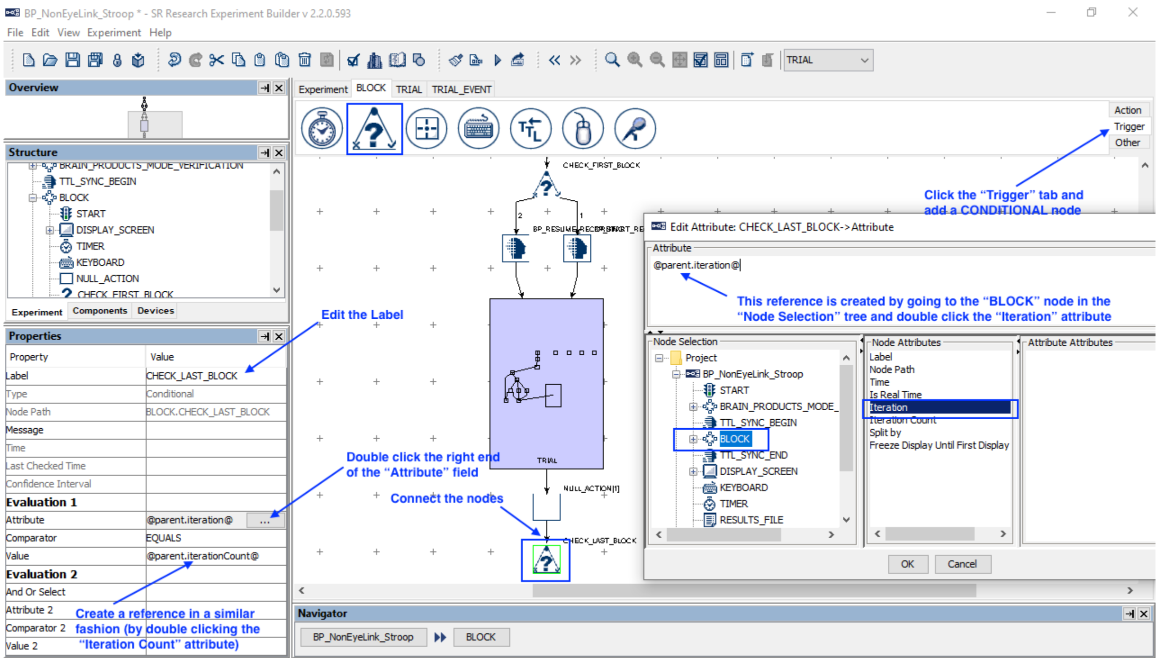
5) Select the “Action” tab of the Component Toolbox and add a BRAIN_PRODUCTS_CONTROL node to the graph. Edit the “Label” and “Message” properties of the node to “BP_RESUME_RECORDING”. Set the “Operation” to “Control Recording”, and then choose “Resume Recording” option from the “Control Recording” drop-down list. Now draw a connection from the left branch of the CHECK_FIRST_BLOCK conditional trigger to BP_RESUME_RECORDING, and from the BP_RESUME_RECORDING node to the TRIAL sequence.



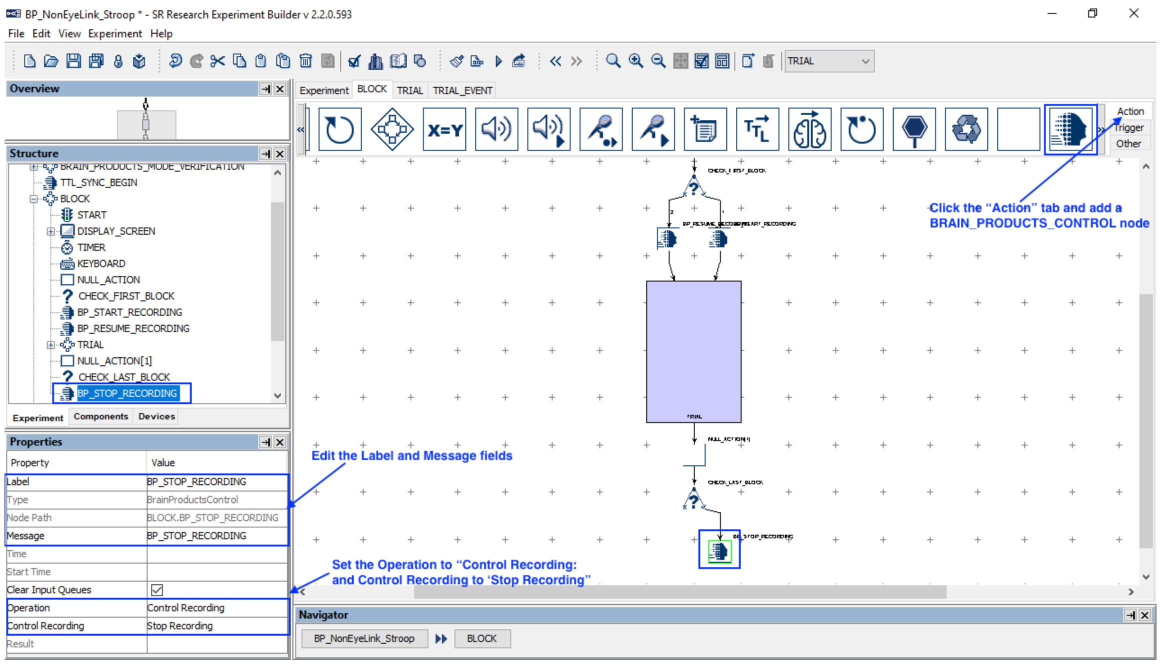
6) We will do similar operations at the end of each block. For the first block, we will pause the EEG recording at the end of the block, and then at the last block, we will stop the EEG recording. Select the “Action” tab of the Component Toolbox, click the NULL action ( ), and drag the node into the graph. This action doesn’t do anything except for making it possible to add a trigger following the TRIAL sequence. Make a connection from the TRIAL sequence to the NULL_ACTION node.

7) Select the “Trigger” tab of the Component Toolbox, click the CONDITIONAL trigger ( ), and drag the node into the graph. Make a connection from the NULL_ACTION node to the CONDITIONAL node. Select the newly added CONDITIOAL trigger. Rename the “Label” as “CHECK_LAST_BLOCK”. Double click the right end of the “Attribute” field. This brings up an attribute editor dialog box. In the “Node Selection” panel on the left, find the BLOCK node and double click the “Iteration” node in the “Node Attributes” panel in the middle. This will automatically create a reference @parent.iteration@. Click the OK button to close the dialog box. Set the “Comparator” to “EQUALS” and

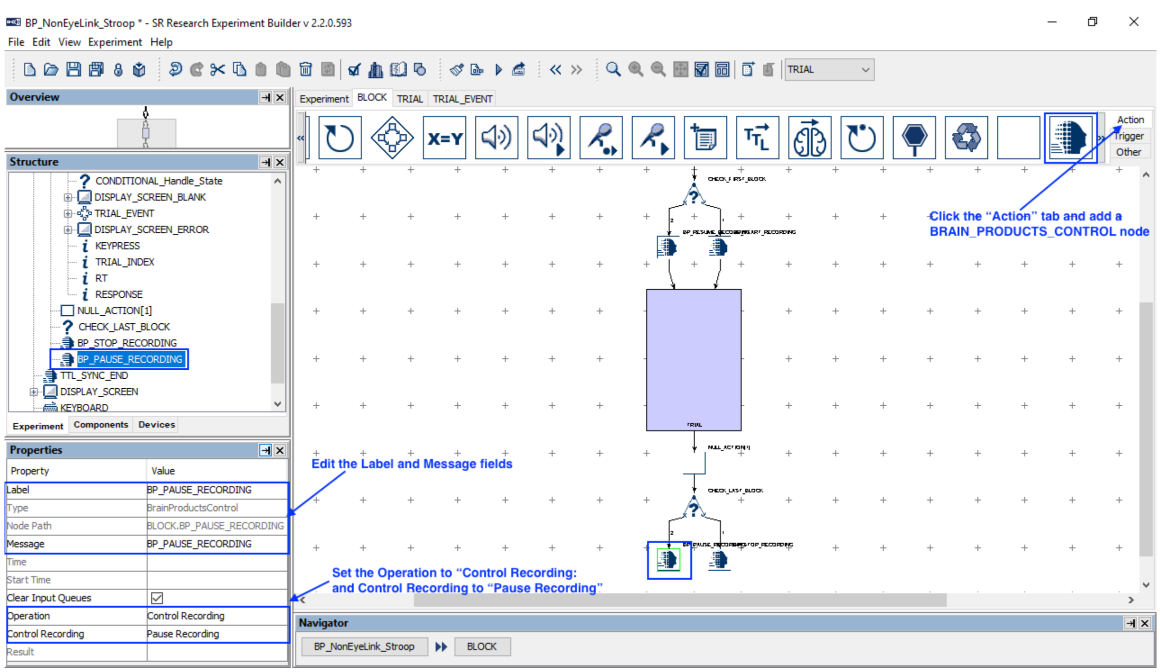
"Value" to @parent.iterationCount@. This conditional trigger is used to check whether the current block is the last block (i.e., total block count).



8) Select the "Action" tab of the Component Toolbox and add a BRAIN_PRODUCTS_CONTROL node to the graph. Edit the "Label" and "Message" properties of the node to BP_STOP_RECORDING. Set the "Operation" to "Control Recording", and then choose "Stop Recording" option from the "Control Recording" drop-down list. Now draw a connection from the right branch of the CHECK_LAST_BLOCK conditional trigger to the BP_STOP_RECORDING node.

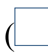


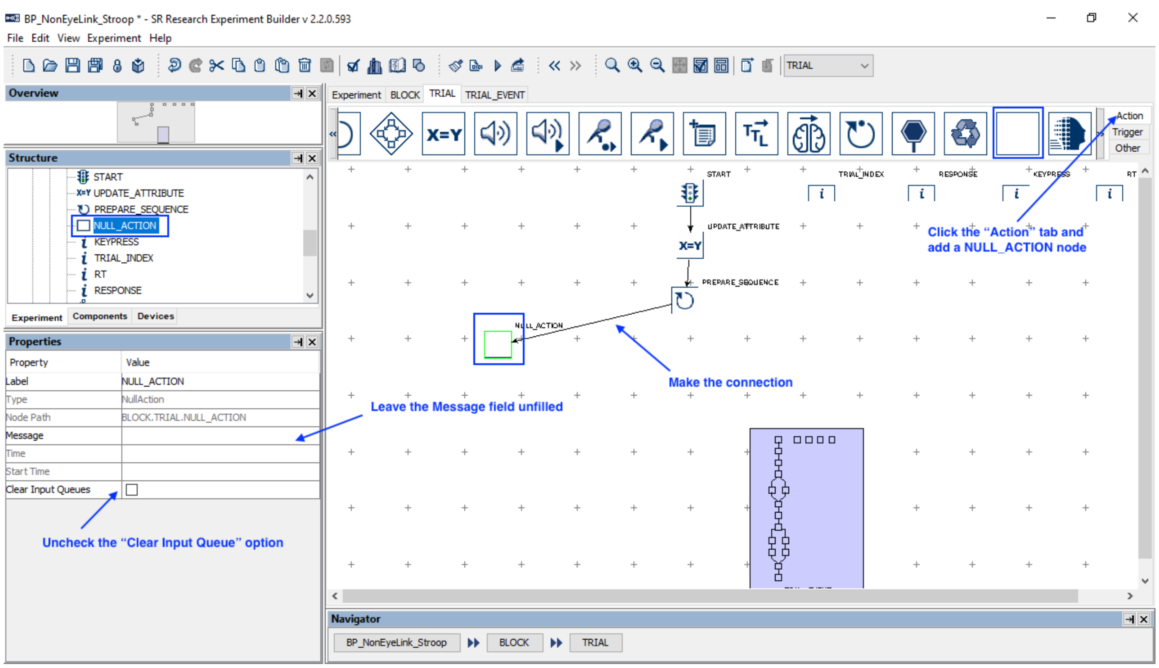
9) Select the "Action" tab of the Component Toolbox and add another BRAIN_PRODUCTS_CONTROL node to the graph. Edit the "Label" and "Message" properties of the node to BP_PAUSE_RECORDING. Set the "Operation" to "Control Recording", and then choose "Pause Recording" option from the "Control Recording" drop-down list. Now draw a connection from the left branch of the CHECK_LAST_BLOCK conditional trigger to the BP_PAUSE_RECORDING node.



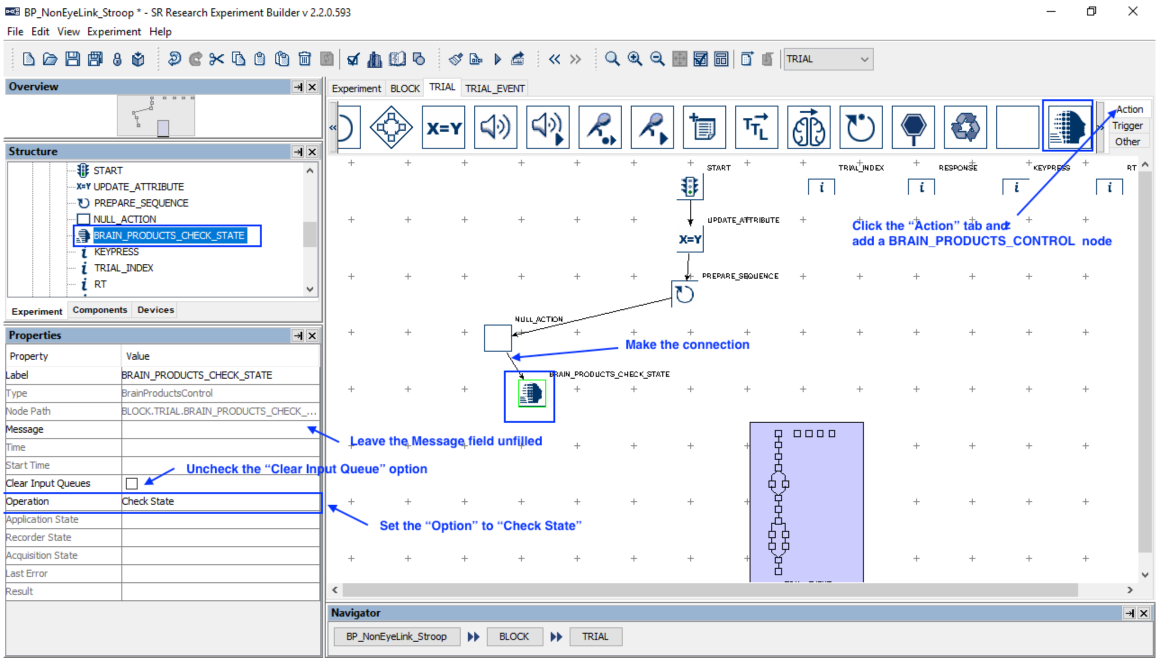
3 Checking the EEG Recording States

One important aspect of the current integration is that Experiment Builder is able to get the states of the EEG system (Recorder, Application, Acqusition) so that experiment flow can be controlled in case of errors. This example illustrates checking the states of the EEG system at the beginning of each trial with a BRAIN_PRODUCTS_CONTROL action. In case of any errors on the EEG side, the experiment can be paused while the states of the EEG systems are still constantly monitored until the experimenter resolves the issues.

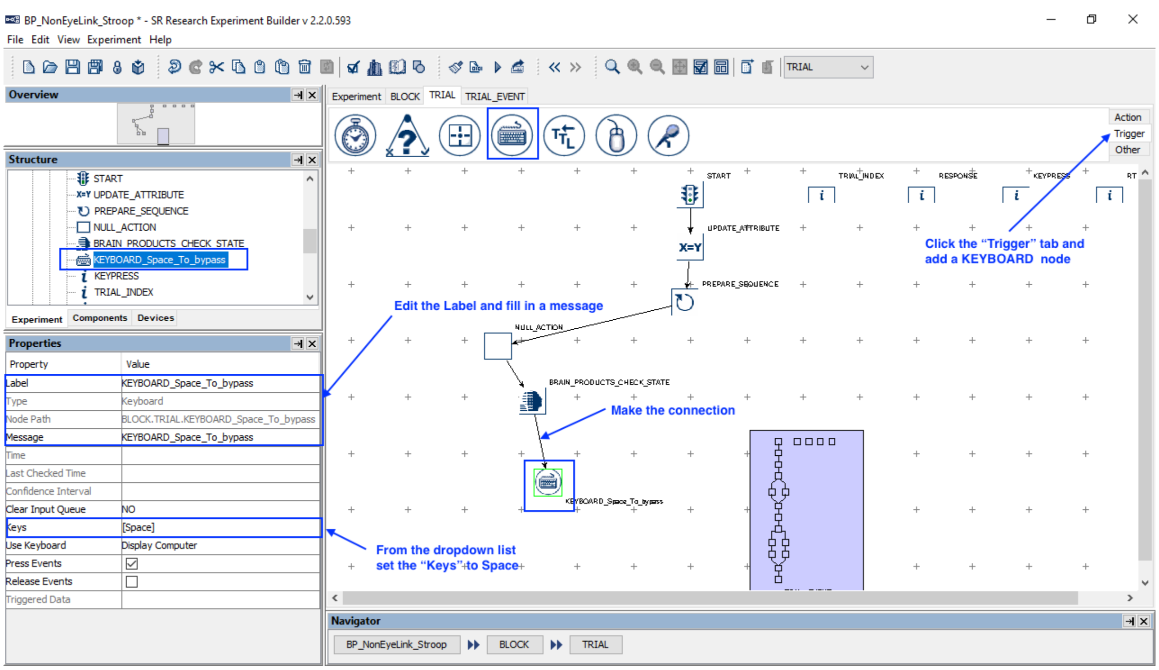
1) Double click the TRIAL sequence.
2) Select the "Action" tab of the Component Toolbox, click the NULL action (), and drag the node into the graph. This action doesn't do anything except for letting the experiment flow loop back in case of EEG errors. Leave the "Message" field unfilled so that the messages will not be sent repeatedly in case of error. The "Clear Input Queues" option of the NULL_ACTION should be unchecked so that key presses will not be cleared before they are actually processed. Connect from the PREPARE_SEQUENCE node to the NULL_ACTION node.



3) Select the "Action" tab of the Component Toolbox and add a BRAIN_PRODUCTS_CONTROL node to the graph. Edit the "Label" of the node to "BRAIN_PRODUCTS_CHECK_STATE". Leave the "Message" field unfilled and uncheck the "Clear Input Queues" option of node. Set the "Operation" to "Check State". Now draw a connection from the NULL_ACTION to BRAIN_PRODUCTS_CHECK_STATE node.

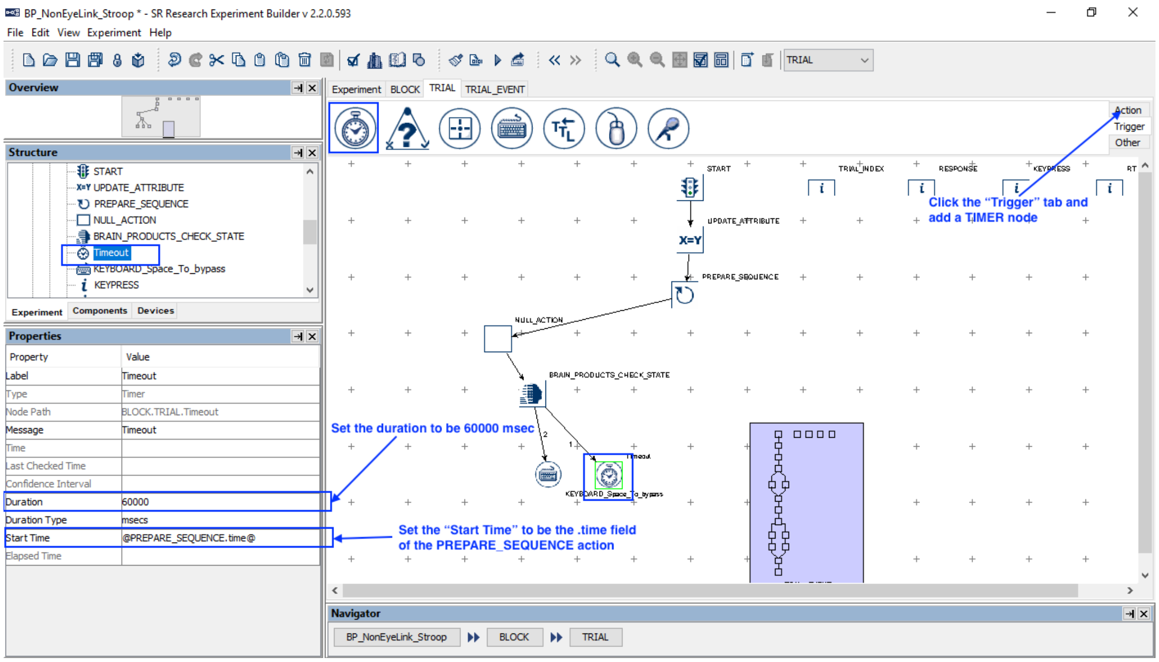


4) Select the "Trigger" tab of the Component Toolbox, click the KEYBOARD trigger, and drag the node into the graph. Fill out the message and set the 'Keys" to [Space]. Draw a connection from the BRAIN_PRODUCTS_CHECK_STATE node to the KEYBOARD trigger node. This allows users to skip the EEG state checking by pressing the spacebar on the display computer.



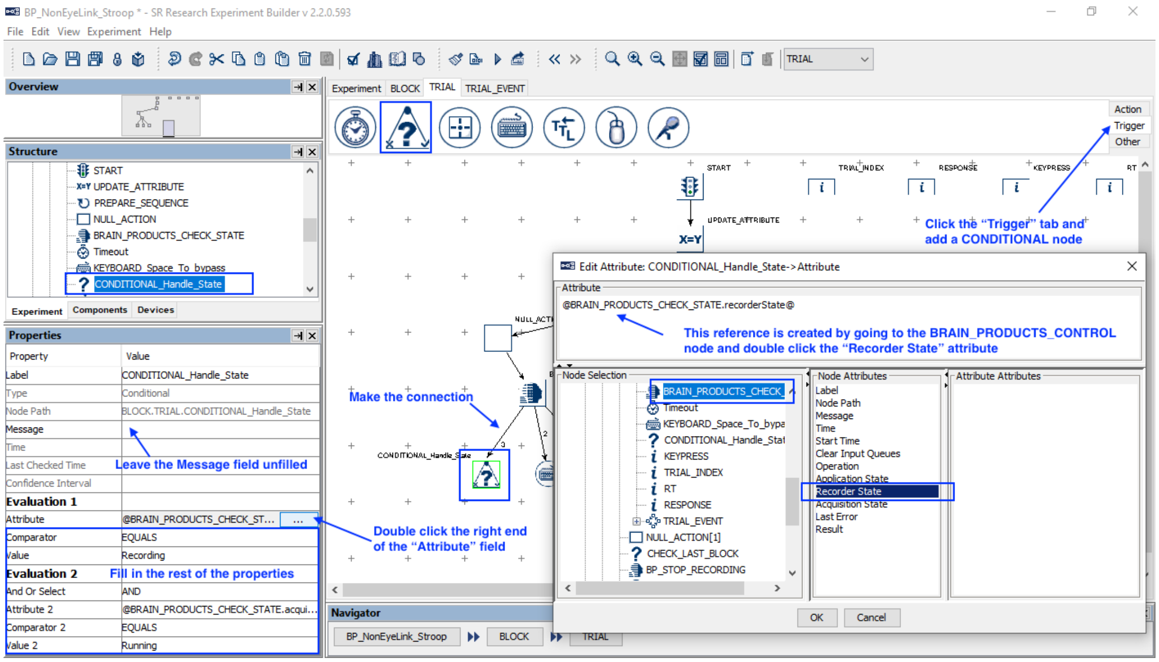
5) Select the "Trigger" tab of the Component Toolbox, click the TIMER trigger, and drag the node into the graph. Fill out the message and set the "Duration" to 60000 (msecs). The Start Time of the TIMER trigger should be

@PREPARE_SEQUENCE.time@. Draw a connection from the BRAIN_PRODUCTS_CHECK_STATE node to the TIMER trigger node.

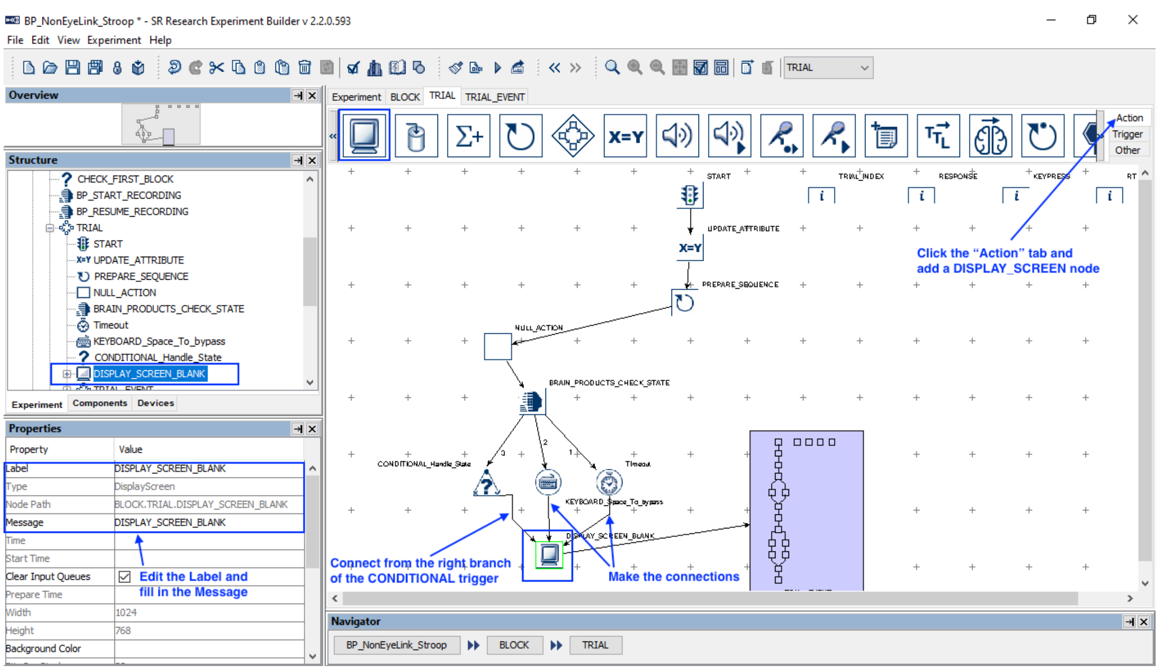


6) Select the "Trigger" tab of the Component Toolbox, click the CONDITIONAL trigger, and drag the node into the graph. Rename the Label as "CONDITIONAL _Handle State". Leave the "Message" field empty. Double click the right end of the "Attribute" field. This brings up an attribute editor dialog box. In the "Node Selection" panel on the left, find the BRAIN_PRODUCTS_CHECK_STATE node and double click the "Recorder State" node in the middle "Node Attributes" panel. This will create a reference @BRAIN_PRODUCTS_CHECK_STATE.recorderState@. Click the OK button to close the dialog box. Set the "Comparator" to "EQUALS" and "Value" to "Recording". This conditional evaluation checks whether the Recorder is running.

   For the properties under "Evaluation 2", set "And Or Select" to AND. Double click the right end of the "Attribute 2" field. In the "Node Selection" panel of theattribute editor dialog box, find the BRAIN_PRODUCTS_CHECK_STATE node and double click the "Acqusition State" node in the middle "Node Attributes" panel. This will create a reference @BRAIN_PRODUCTS_CHECK_STATE.acquisitionState@. Click the OK button to close the dialog box. Set the "Comparator" to "EQUALS" and "Value" to "Running". This conditional evaluation checks whether the acqusition is running. Draw a connection from the BRAIN_PRODUCTS_CHECK_STATE node to the CONDITIONAL trigger node.

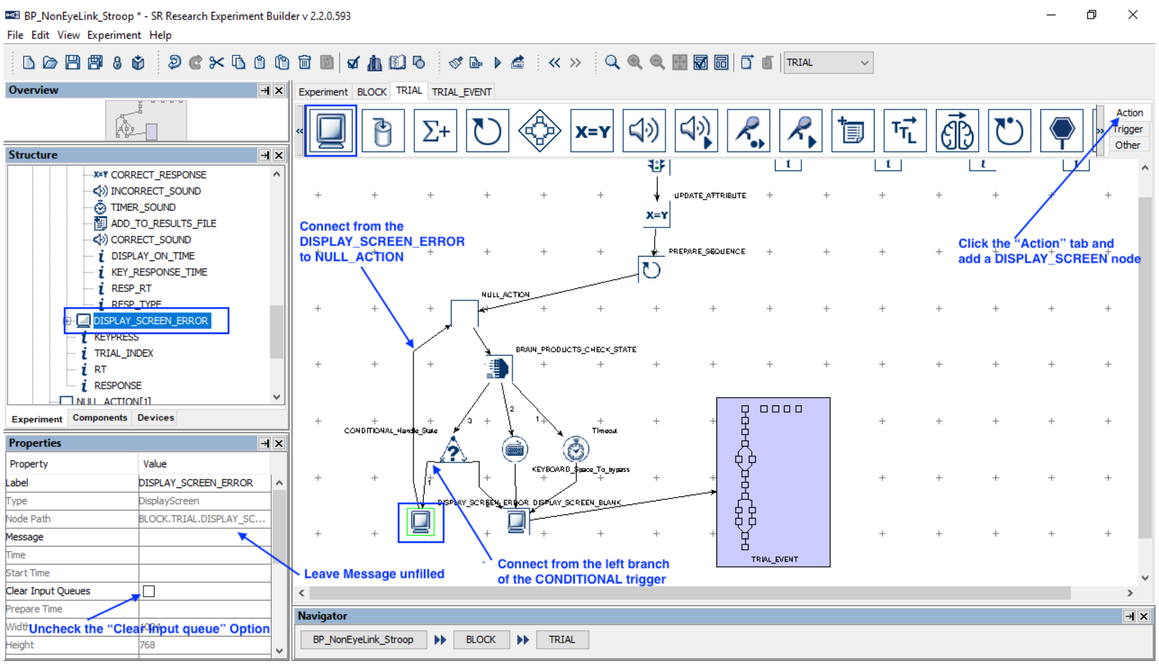


7) Select the "Action" tab of the Component Toolbox and drag a "DISPLAY_SCREEN" node into the experiment graph. Edit the label of the node as "DISPLAY_SCREEN_BLANK" and fill out a message. You don't need to do anything to the action as it will be simply showing a blank screen. Make a connection from the TIMER, KEYBOARD trigger nodes, as well as the right branch of the CONDITIONAL_Handle_State trigger.

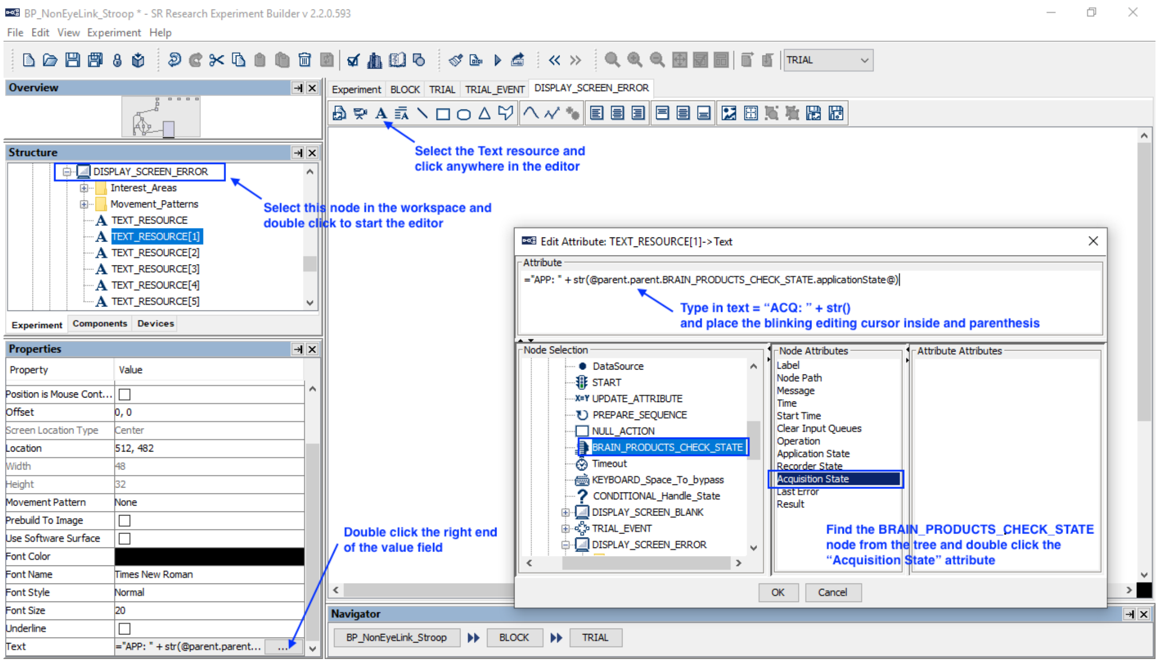


8) Select the "Action" tab of the Component Toolbox and drag a "DISPLAY_SCREEN" node into the experiment graph. Edit the label of the node

as “DISPLAY_SCREEN_ERROR” and leave the message field empty so not to flood the link with repetitive messages. The “Clear Input Queues” option of the node should be unchecked. Make a connection from the left branch of the CONDITIONAL_Handle_State trigger. Make a connection from the DISPLAY_SCREEN_ERROR node to the NULL_ACTION node.



9) Select the newly added DISPLAY_SCREEN_ERROR node in the graph. Double click to start the Screen Builder. Click the text resource on the toolbar and then click anywhere on the screen editor. This will add a text resource. Click the right end of the “Text” field. In the Edit Attribute editor box, type in the following in the top “Attribute” box: =“ACQ: ” + str()
Please note an “=” sign is added at the beginning of the text to indicate this is an equation. While leaving the blinking editing cursor inside the parenthesis, find the BRAIN_PRODUCTS_CHECK_STATE node in the “Node Selection” treeview and double click the “Acquisition State” node in the middle “Node Attributes” panel. This will create a reference @parent.parent.BRAIN_PRODUCTS_CHECK_STATE.acqusitionState@ for the equation in the Attribute box.



10) Repeat this process to create a few other text resources to report the states returned from the BRAIN_PRODUCTS_CHECK_STATE node:

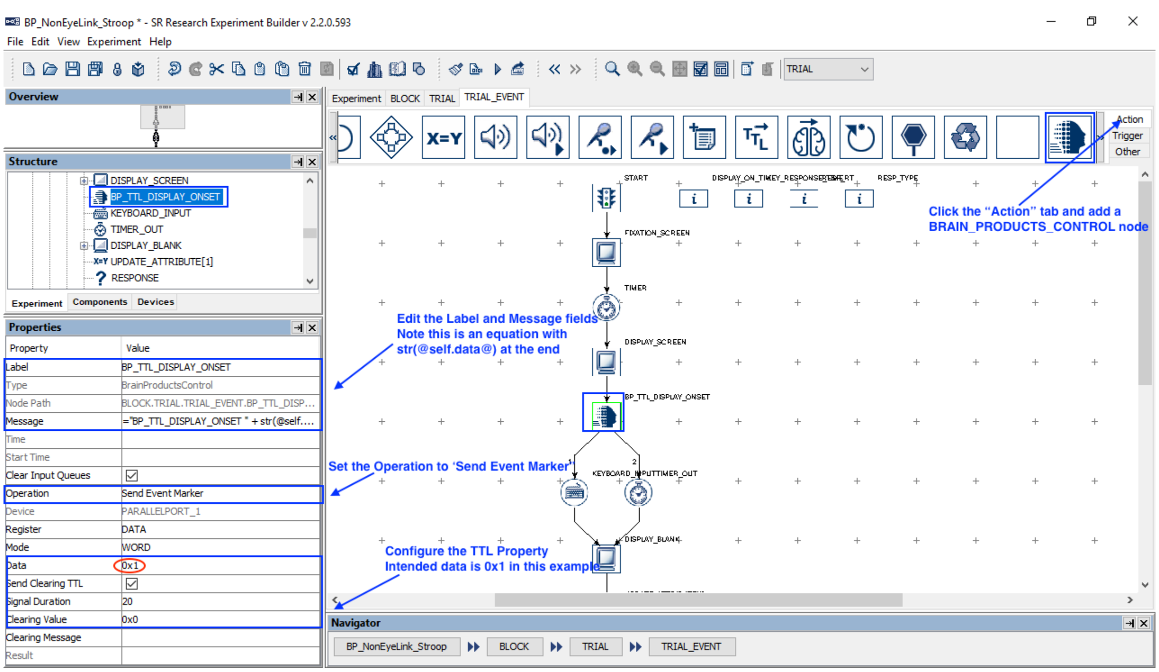
="ACQ: " + str(@parent.parent.BRAIN_PRODUCTS_CHECK_STATE.acquisitionState@)
="REC: " + str(@parent.parent.BRAIN_PRODUCTS_CHECK_STATE.recorderState@)
="APP: " + str(@parent.parent.BRAIN_PRODUCTS_CHECK_STATE.applicationState@)
="ERR: " + str(@parent.parent.BRAIN_PRODUCTS_CHECK_STATE.lastError@)

4 Sending Event Markers to EEG Recordings

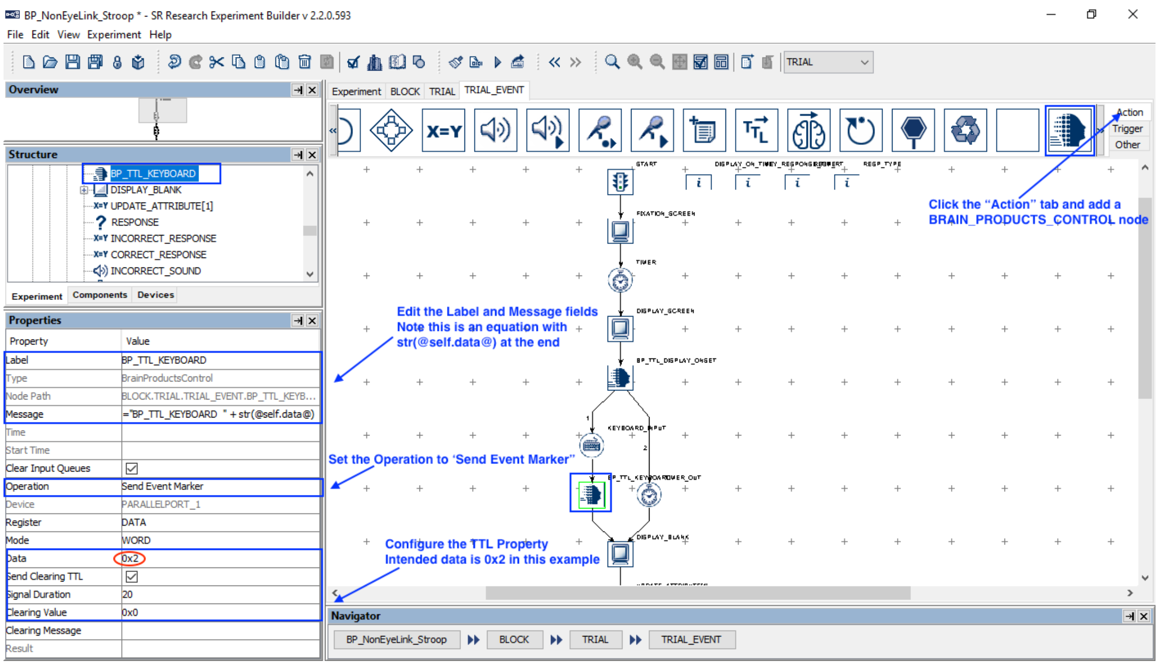
To make sure events recorded in Experiment Builder are synchronized with the EEG data collected in BrainVision Recorder, users can send TTL signals from Experiment Builder to mark the critical events in the experiment such as display and audio stimulus onsets, participant responses, etc. In this example, we will illustrate how to use a BRAIN_PRODUCTS_CONTROL action to mark the onset of the DISPLAY_SCREEN and send the participant's response data to the EEG data stream.

1) Double click the innermost TRIAL_EVENT sequence to show events in the trial.
2) Select the "Action" tab of the Component Toolbox and add a BRAIN_PRODUCTS_CONTROL action. Set the message of the action to '="BP_TTL_DISPLAY_ONSET" + str(@self.data@)' (note that the expression must begin with an "="). The reference str(@self.data@) will record the TTL value in the message text. Set the "Operation" to "Send Event Marker". Set the "Data" to 0x1, signal duration to 20 ms, and clearing value to 0x0. Draw a connection from the DISPLAY_SCREEN action to the newly added BP_TTL_DISPLAY_ONSET node. From the BP_TTL_DISPLAY_ONSET node, draw connections to KEYBOARD_INPUT, and TIMER nodes.

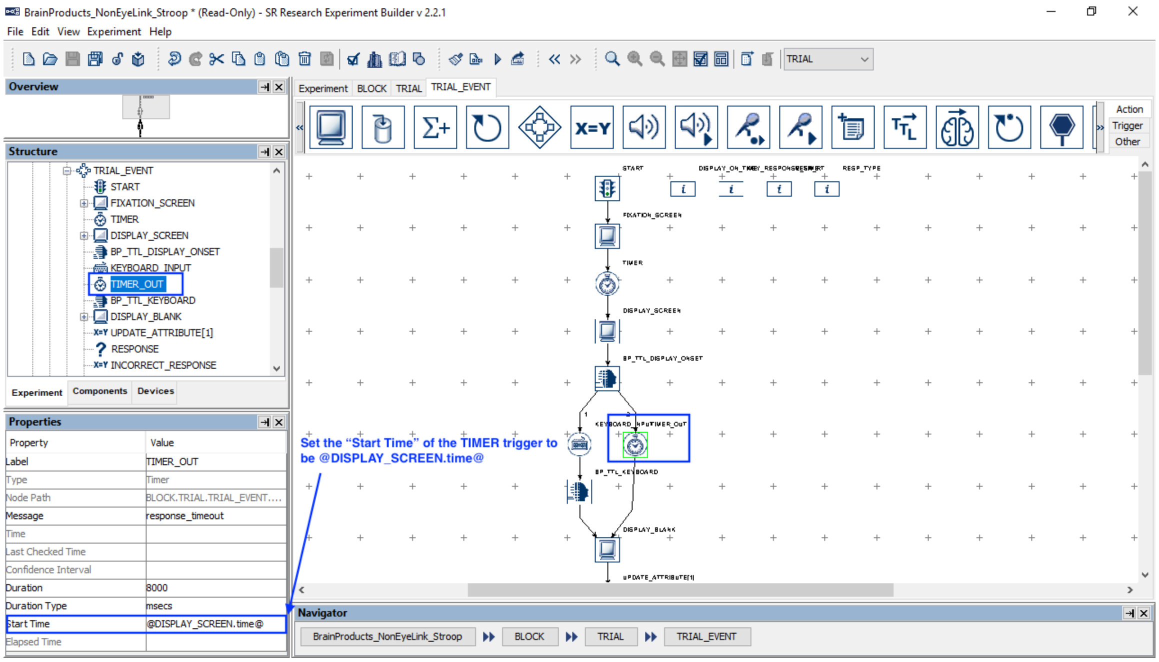
   Important! For proper data alignment between the behavioral data and EEG recordings, please make sure the "Message" field of the BRAIN_PRODUCTS_CONTROL action contains a uniquely identifiable text (with the TTL value at the end of the text). Please also make sure the BRAIN_PRODUCTS_CONTROL node is placed immediately after the event you want to mark in the EEG data.



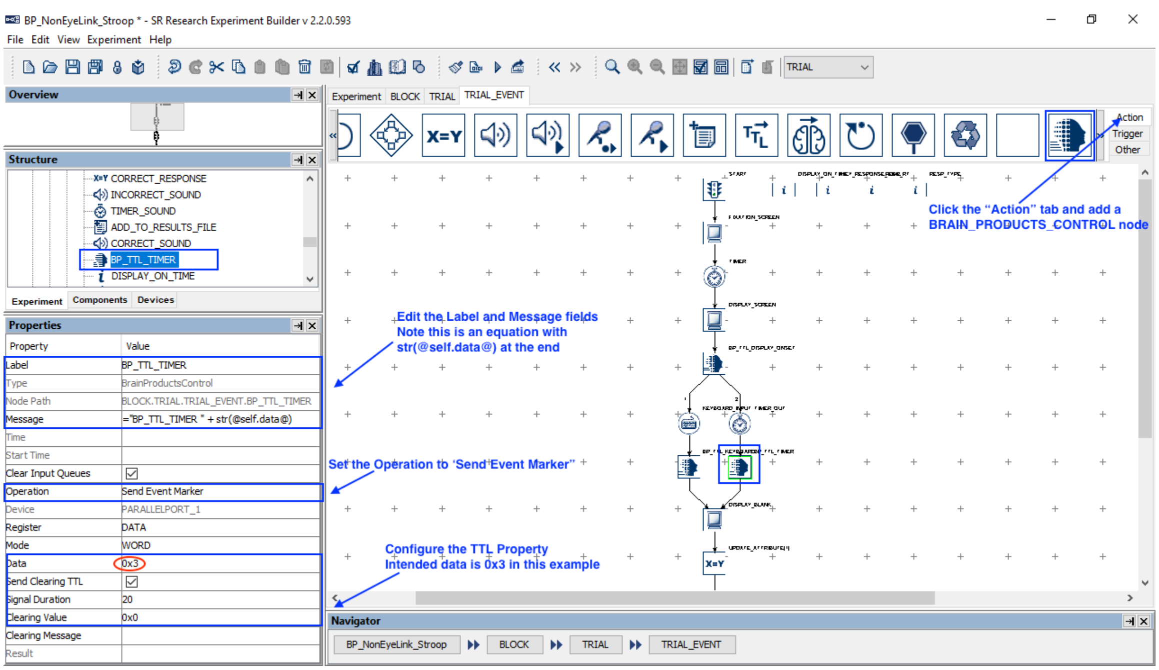
3) Next, send a TTL signal following each of the possible response methods. Let's start with the keyboard response. Select the "Action" tab of the Component Toolbox and add a BRAIN_PRODUCTS_CONTROL action. Set the message of the action to ="BP_TTL_KEYBOARD" + str(@self.data@). Set the "Operation" to "Send Event Marker". Set the "Data" to 0x2, signal duration to 20 ms, and clearing value to 0x0. Draw a connection from the KEYBOARD_INPUT to BP_TTL_KEYBOARD.



4) For the TIMER trigger that follows the BP_TTL_DISPLAY_ONSET node, set the Start Time to be @DISPLAY_SCREEN.time@ instead of the default value of 0. With this change, the elapse time of the TIMER trigger starts at the onset of the DISPLAY_SCREEN action instead of the return of the BIOMETRIC_TTL action (after the clearing signal is sent).

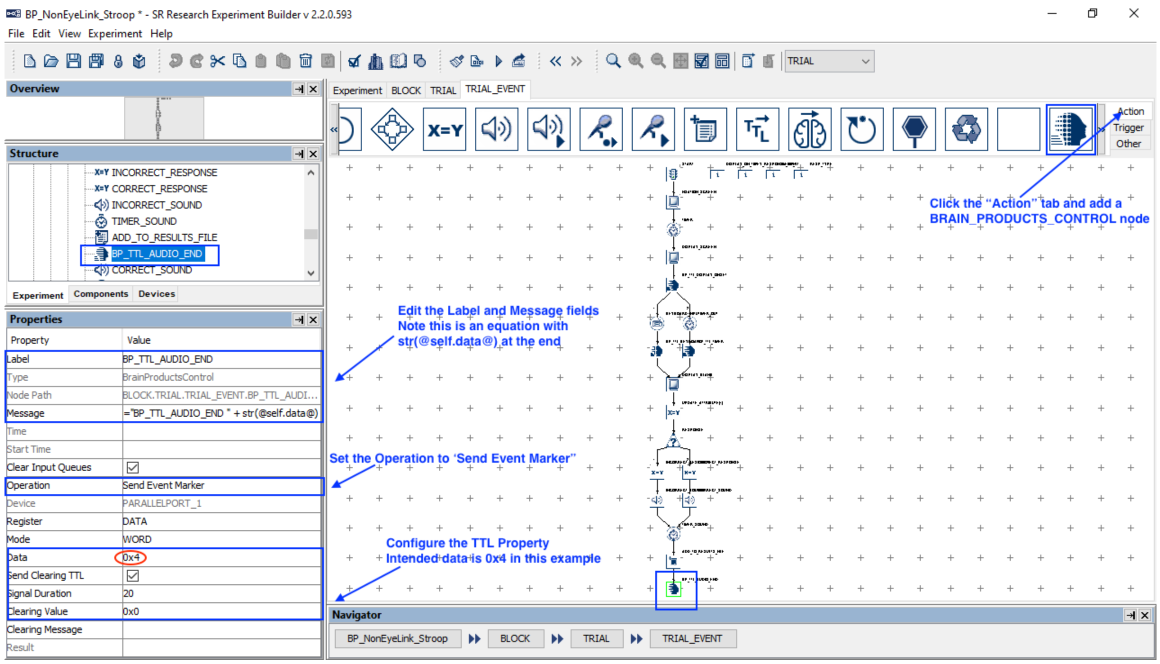


5) Now select the "Action" tab of the Component Toolbox and add a BRAIN_PRODUCTS_CONTROL action. Set the message of the action to ="BP_TTL_TIMER" + str(@self.data@). Set the "Operation" to "Send Event Marker". Set the "Data" to 0x3, signal duration to 20 ms, and clearing value to 0x0. Draw a connection from the TIMER to BP_TTL_TIMER.



6) Finally, send a TTL when the feedback audio is finished playing. Select the "Action" tab of the Component Toolbox and add a BRAIN_PRODUCTS_CONTROL action. Set the message of the action to

="BP_TTL_AUDIO_END" + str(@self.data@). Set the "Operation" to "Send Event Marker". Set the "Data" to 0x4, signal duration to 20 ms, and clearing value to 0x0. Draw a connection from the ADD_TO_RESULTS_FILE to BP_TTL_AUDIO_END.



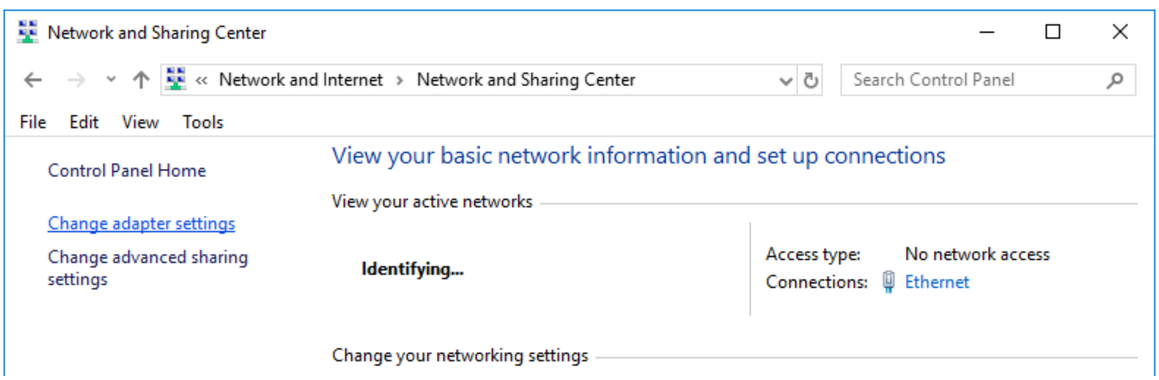
5 Configuring Network Settings for the Computers

The Display PC that runs Experiment Builder and the BrainVision Recorder PC should be connected through a network cable (a crossover network cable is preferred; a straight-through cable will be fine if a hub or switch is used). This example assumes the IP address is set to 100.1.1.2 for the Display Computer and 100.1.1.3 for the BrainVision Recorder computer. Users are encouraged to use this default IP range. The following sections discuss how to configure the IP addresses of the Display PC and BrainVision Recorder computer.

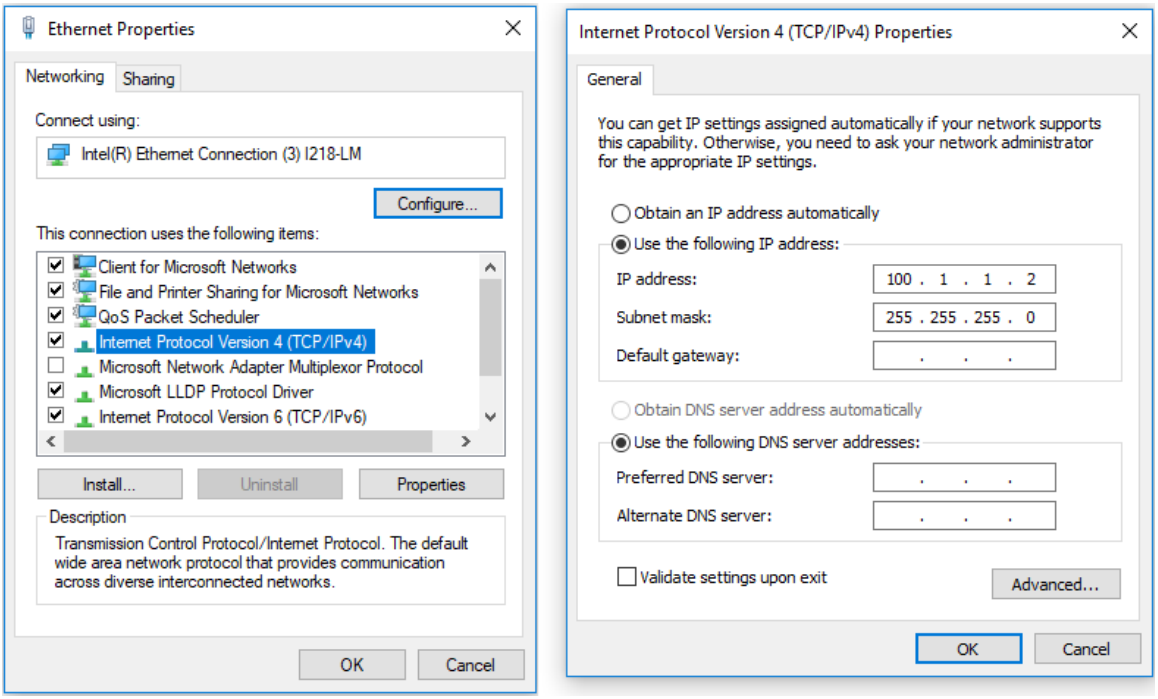
5.1 Configuring the Experiment Builder Computer IP Address

The following instructions are based on Windows 10; instructions for other Windows operating systems may vary slightly.

1) From the Start menu, select “Windows System -> Control Panel”.
2) Click on the “Network and Internet” icon, and then select the “Network and Sharing Center” icon. In the following Screen, choose “Change adapter settings” icon on the left side panel (see the Figure below).



3) Check the list of installed components to make sure a network card is detected. If not, install the driver for the network card.
4) Double click on the network card icon that represents the network card that will be connected to the EyeLink Host PC.
5) Select the “Properties” button.
6) Select the “Internet Protocol Version 4 (TCP/IPv4)” and then click on the “Properties” button (see the figure below).

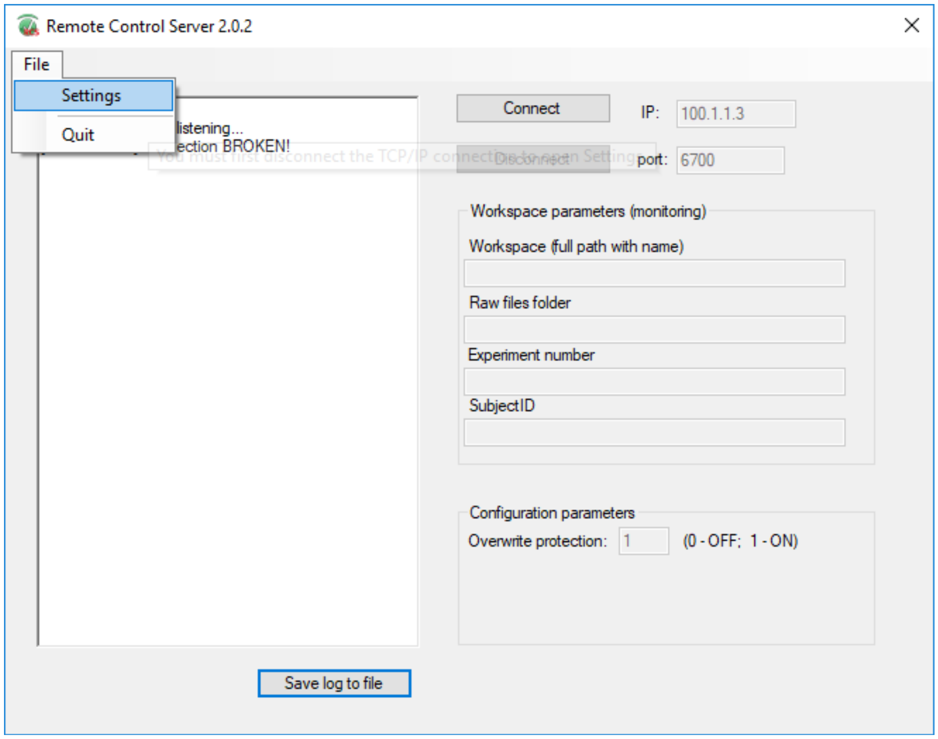


7) Select the "Use the following IP address" radio button. Enter the IP address "100.1.1.2" for the display computer that runs Experiment Builder. The last digit of the IP address can be increased as necessary to account for other computers on the network. Enter the subnet mask of "255.255.255.0". Leave the default gateway and other settings blank. If you are using a different IP address for the BrainVision Recorder computer, please make sure you update the "IP Address" of the Brain Products device (see section 1.2) in Experiment Builder.
8) Click on "OK" to return to the Properties dialog. Click "OK" again to save your changes. Click "Close" to exit from the network card dialog.

5.2 Configuring the BrainVision Recorder Computer IP Address and Port

Users should follow the same instructions for configuring the IP address and subnet mask for the computer that runs the BrainVision Recorder, except that the IP address should be "100.1.1.3". User should additionally tweak the IP configuration in the Remote Control Server.

1) Start the BrainVision Remote Control Server application. Click "File -> settings"



2) In the following settings dialog box, enter "IP" as 100.1.1.3 and Port as 6700.