

# Adding Event Markers to an E-Prime Experiment

---

This tutorial shows you how to add ClearView PackageCall objects to an existing experiment in order to perform scene-based analyses in Tobii Studio. This tutorial picks up where Tutorial 4 left off in the **E-Prime Extensions for Tobii Pro 3.0 User Manual**. If you have completed Tutorial 4, then you will begin this tutorial by opening your finished Tutorial 4 experiment. If you have not completed Tutorial 4 then you will begin by opening the paradigm, TETFixedPositionAOI.es3. Details are provided in Step 1.

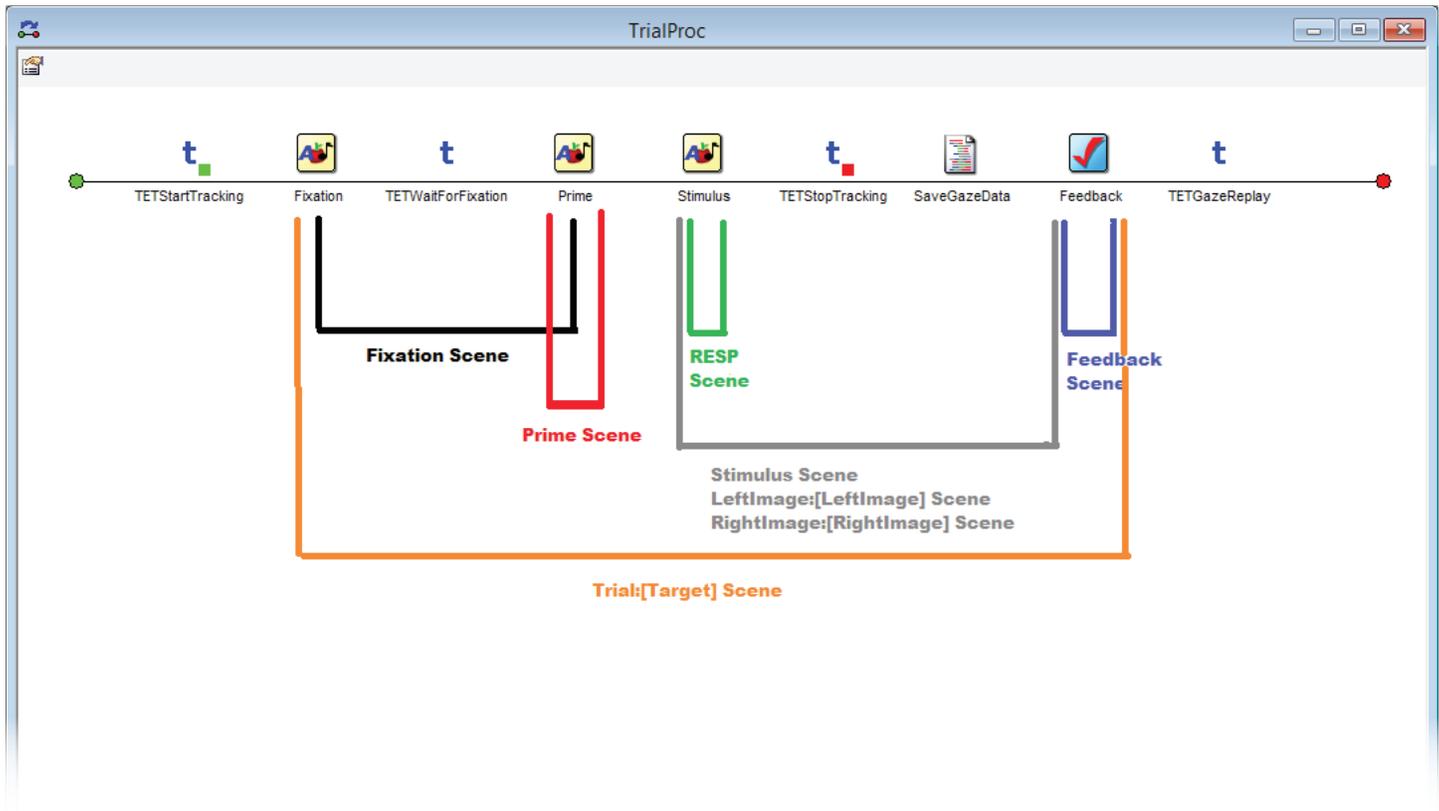
## Summary:

Tutorials 1 through 4 focused on the functionality supported by the TET PackageFile, which is one of the two package files that are provided with EET. To review, the TET PackageFile routines support communication with the TET Server. Specifically, these routines support the capture and replay of gaze data at the conclusion of each trial, along with the creation of the .gazedata file that can be examined in Excel after the completion of the experiment. When working with the TET PackageFile, E-Prime is the application that is controlling and coordinating the experiment events. The experiment session begins by launching the experiment from E-Studio or with E-Run. During the experiment, E-Prime requests eye tracking information from the Tobii monitor, E-Prime script is called to present a replay of the gaze data for the trial that was just completed, and E-Prime combines the experiment data it collects (e.g., trial number, response time) with the Tobii eye tracking data and writes the combined information out to the .gazedata file. Lastly, once the experiment session is complete, the .gazedata file can be analyzed in Excel.

This tutorial introduces the functionality supported by the second package file in EET, the ClearView (CV) PackageFile. The CV PackageFile is designed for users who need to perform more visually-sophisticated eye tracking data analysis in Tobii Studio than can be performed in Excel on the .gazedata file. Specifically, Tobii Studio supports analysis by scenes, where a scene is defined as a pair of scene markers that is inserted into the data stream. Each scene has a name, such as “Fixation”, and a Scene Start and Scene Stop marker. The CV PackageFile supports the creation of scenes by providing customized Task Events that send the scene markers to Tobii Studio. By using the CV PackageCalls and E-Prime's Task Events, an E-Prime experiment can be customized to instruct Tobii Studio to start a Scene, end a Scene, and to log important events within the trial. Incorporating this functionality into an existing E-Prime experiment involves adding CV PackageCalls to the E-Prime experiment structure and adding Task Events to the critical E-Prime objects that correspond to the start and end of Scenes. This process is demonstrated in this tutorial.

The Task Events feature in E-Prime 3.0 is a powerful tool, which enables several tasks to execute when several events occur. Examples of tasks that can be defined include writing a value to a communication port. Examples of events that can be identified as the triggering event include when an E-Prime display object begins to execute. The Task Events feature is used extensively throughout this tutorial to send SceneStart and SceneStop markers to Tobii Studio for the various scenes (such as Fixation, Trial) that are defined for analysis in Tobii Studio. Users who are not familiar with E-Prime Task Events are urged to review Section 3.1 Task Events of the New Features/Reference Guide prior to completing this tutorial.

The figure on the following page illustrates the scenes that will be defined in this tutorial. Note that some scenes, such as the Feedback scene, span on the duration of single object, while other scenes, such as Trial, span over multiple objects.



E-Prime and the Tobii Studio software run on different computers, which allow each application to complete their designated function. However, this requires that the computers communicate with one another. In the case of the two-computer setup using T-Series eye tracker, E-Prime and Tobii Studio both communicate with the Tobii Eye Tracker server via a network switch. PackageCalls and/or Task Events are added at the appropriate place in the E-Prime experiment, and the required parameter values are then specified and sent to Tobii Studio. The data from E-Prime is sent with the Ethernet port via the switch to Tobii Studio in under 10 ms. The switch connection allows for the data to be transmitted between the two software programs, but leaves Tobii Studio blind to what is being presented on screen by the E-Prime software.

In order for the scenes and segments displayed by E-Prime to be properly analyzed by Tobii Studio, the Tobii Studio information must first pass through a video capture card. This allows for a video of the experiment to be saved in Tobii Studio, leading to more robust and meaningful Tobii Studio data. Meanwhile, the visual aspects of the E-Prime experiment from are sent to a splitter. The splitter sends the signal to both the monitor and to the video capture card located on the Tobii Studio machine (typical delay is ~40 ms, but is dependent on the specific hardware configuration), and Tobii Studio collects the visual input via an external video device. For more information about the two computer set up see **Appendix B: Hardware Configurations** in the **E-Prime Extensions for Tobii Pro 3.0 User Manual**.

Tobii Studio converts the E-Prime display into a Tobii Studio video; it can then overlay the gaze information from Tobii monitor onto the video. In order to do this, Tobii Studio must first be started so that the entire E-Prime experiment is included in the Tobii Studio video.

**Goal:**

By the end of this tutorial, you will have created a basic ClearView-enabled E-Prime paradigm that uses Task Events to send event markers to Tobii Studio for scene-based analysis. This tutorial creates eight scenes. In practice, this is likely to be many more scenes than you would typically define for a particular experiment. However, throughout the tutorial a variety of techniques are illustrated, including how to define multiple Task Events on a single object, performing multiple “tasks” (i.e. defining multiple scenes) within a single Task Event, creating scenes that span multiple display objects, and creating scenes that both start and stop on the same object. Each scene requires both a SceneStart and SceneStop, resulting in sixteen Task Events in this tutorial. Refer to the table below to keep track of the scenes:

Scene Name	Tutorial Task, SceneStart	E-Prime Object/ Event to trigger SceneStart	Tutorial Task, SceneStop	E-Prime Object/ Event to trigger SceneStop
Fixation	Tasks 10-11	Fixation.OnsetTime	Tasks 13-14	Prime.OnsetTime
Trial:[Target]	Task 12	Fixation.OnsetTime	Tasks 29-30	Feedback.OffsetTime
Prime	Tasks 15-16	Prime.OnsetTime	Tasks 17-18	Stimulus.OnsetTime
Stimulus	Tasks 19-20	Stimulus.OnsetTime	Tasks 25-26	Feedback.OnsetTime
LeftImage:[LeftImage]	Tasks 19-20	Stimulus.OnsetTime	Task 27	Feedback.OnsetTime
RightImage:[RightImage]	Tasks 19-20	Stimulus.OnsetTime	Task 27	Feedback.OnsetTime
RESP	Tasks 19-20	Stimulus.OnsetTime	Tasks 21-22	Stimulus.Keyboard(1).MaxCountReached
			Tasks 23-24	Stimulus.Keyboard(1).TimeLimitReached
Feedback	Task 27	Feedback.OnsetTime	Tasks 28-29	Feedback.OffsetTime

**Overview of Tasks:**

- Rename TETFixedPositionAOI.es3 as TETFixedPositionAOIMarkers.es3.
- Add the CV package file.
- Add the CV device to the Experiment Property Pages and edit the devices properties to enable communication between E-Prime and Tobii Studio.
- Add the CV PackageCall to initialize the CV package file.
- Add the CV PackageCalls to start/stop the communication between E-Prime and ClearView.
- Add the CV PackageCalls to start/stop recording the experiment to be viewed in Tobii Studio.
- Add and Edit the Target Attribute to the TrialList.
- Add the CV Task Events to mark the start of the scene(s) that correspond to the Trial Object, Fixation Object, Prime Object, Stimulus Object as well as LeftImage, RightImage, and Response. Use the table above as a guide to adding the aforementioned scene markers.

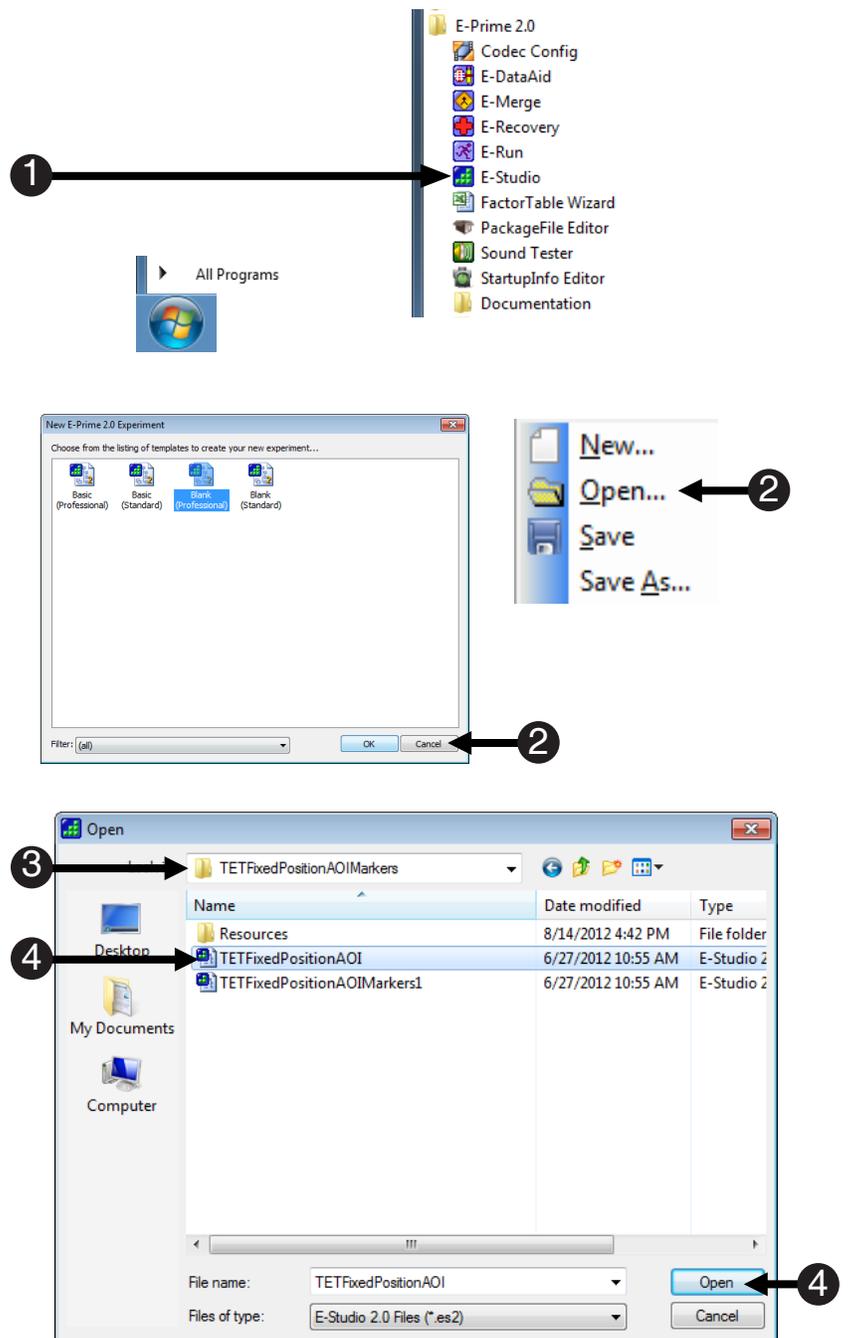
**Estimated Time:** 10-15 minutes

# Task 1: Open the TETFixedPositionAOI.es3 Experiment in E-Studio

Locate the E-Studio icon in the Start > Programs > E-Prime menu and launch the application by double clicking on it. Load the TETFixedPositionAOI.es3 sample experiment.

Open the E-Studio application, navigate to ...My Experiments\Tobii\Tutorials\CV\TETFixedPositionAOIMarkers, and load the TETFixedPositionAOI.es3 experiment. This experiment will act as a template on which we will place the ClearView PackageCallObjects and Task Events.

- 1) **Click** on the Windows **Start** menu, **select All Programs**, and then **select E-Prime**. From the menu, **click** on **E-Studio** to launch the application.
- 2) **Click** the **Cancel** button. **Select File > Open**.
- 3) **Navigate** to the “<My Experiments>\Tobii\Tutorials\CV\TETFixedPositionAOIMarkers” folder.
- 4) **Select** the **TETFixedPositionAOI.es3** file and then **click** the **Open** button to **load** the **paradigm** into E-Studio.

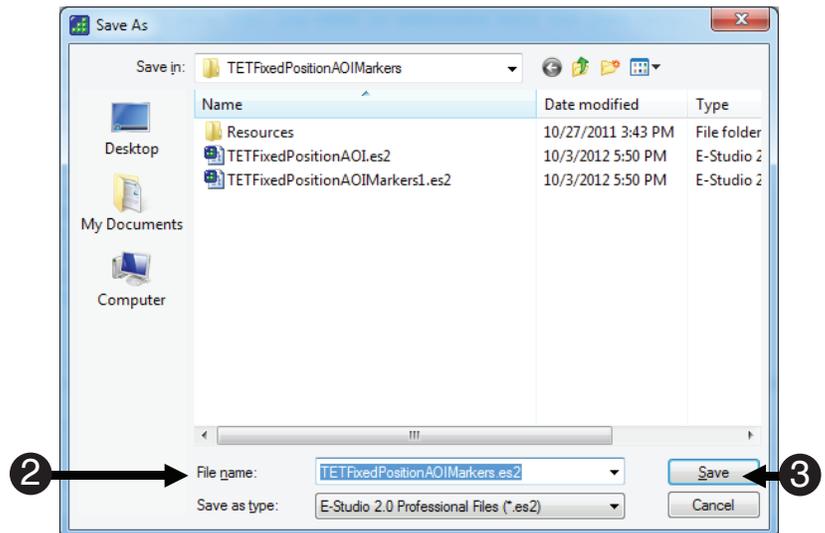
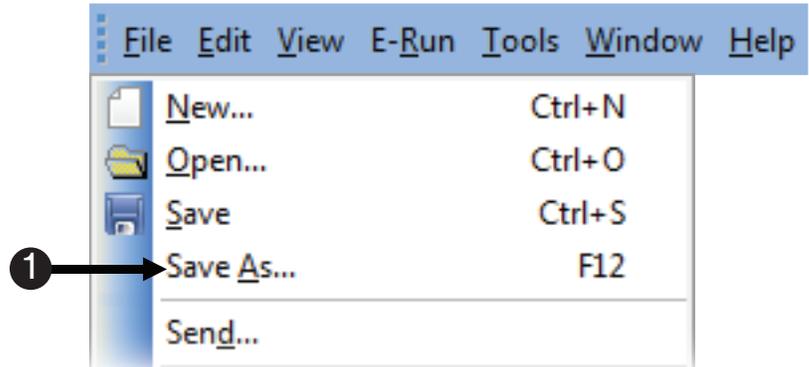


## Task 2: Save the Experiment Under a New Name

Save the *TETFixedPositionAOI.es3* experiment in the same folder under the new name “*TETFixedPositionAOIMarkers.es3*.”

Rename the experiment, but be sure to save it in the same folder (“...\\My Experiments\\Tobii\\Tutorials\\CV\\TETFixedPositionAOIMarkers”) so that any resources within the experiment will remain valid and can be reused.

- 1) **Select File > Save As...** from the application menu bar.
- 2) **Type** *TETFixedPositionAOIMarkers.es3* as the new name in the **File name** field.
- 3) **Click** the **Save** button.

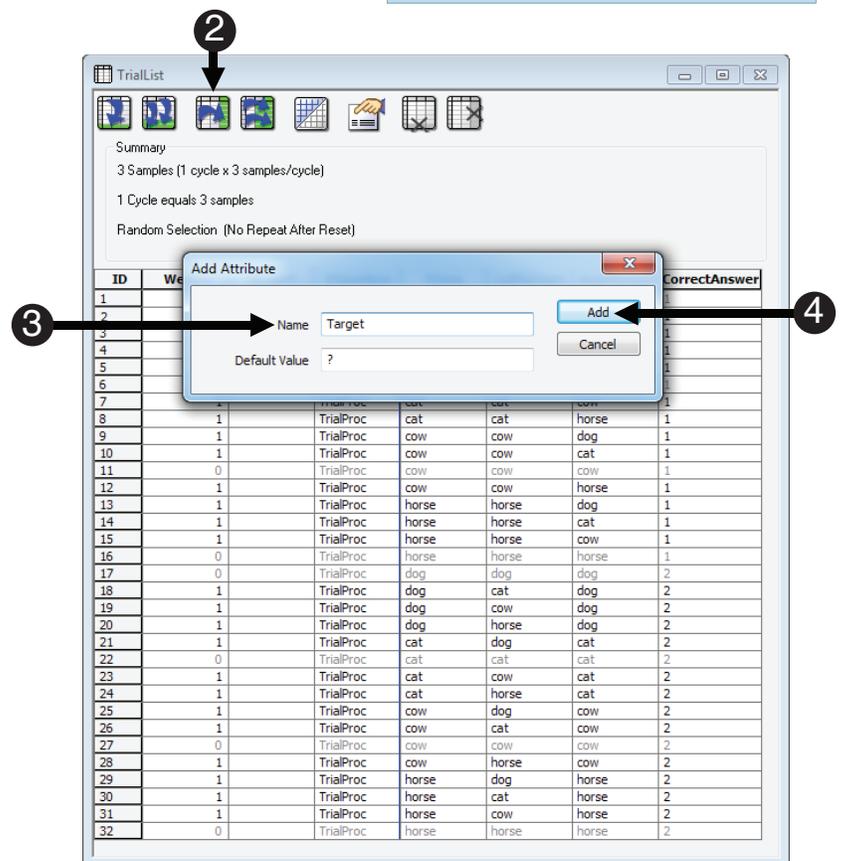
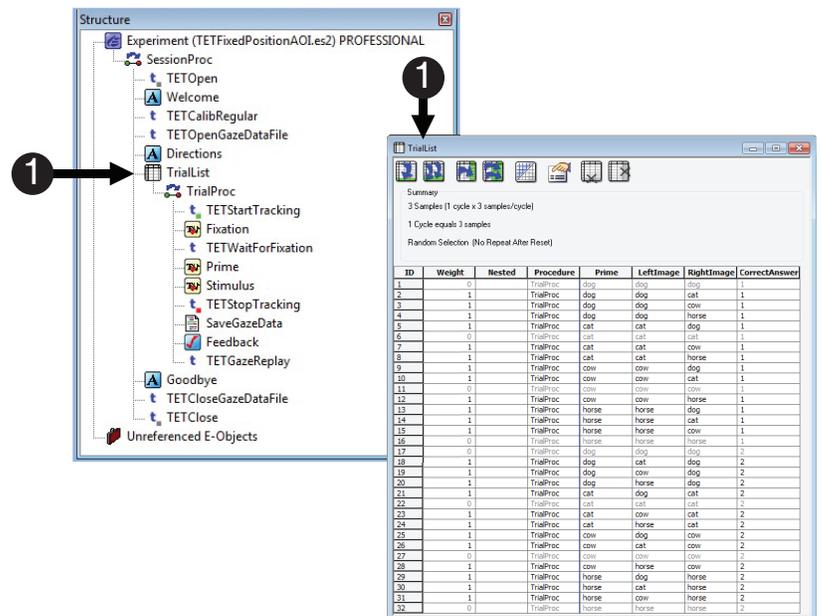


## Task 3: Add an Attribute to the TrialList and name it Target

Add an Attribute named "Target" to the TrialList.

Currently, the TETFixedPositionAOIMarkers.es3 experiment does not explicitly encode the AOI for each trial (left or right). This information would be useful to have when analyzing the gaze data in Tobii Studio. In order to send this information to Tobii Studio, it must first be added to the experiment. In this step, we add an attribute to the TrialList and name it Target. The following task assigns a value to the attribute for each exemplar. Later in this tutorial, we send the value of the Target attribute to Tobii Studio.

- 1) **Double click** the TrialList to open it in the workspace.
- 2) **Click** the **Add Attribute** button.
- 3) **Name** the Attribute "Target."
- 4) **Click** the **Add** button to **accept** the changes.



## Task 4: Edit the Target Attribute

Edit the Target Attribute to identify the critical AOI.

The Target Attribute will be used in Tobii Studio to identify where the target image was displayed relative to the fixation point. The default value assigned to the Target Attribute in Task 3 ("?") is changed in this task to either "Left" or "Right". Although the CorrectAnswer attribute could be used to infer the critical AOI, it does not encode this information in a descriptive manner. It is more useful to have the location of the target picture sent to Tobii Studio ("Left" or "Right") in order to group each condition into similar Scenes rather than sending the CorrectAnswer Attribute of "1" or "2".

- 1) **Click** in the **first cell** under **"Target"** to **highlight** the cell with a **question mark**. **Type "Left"** and **hit Enter** to accept the change.
- 2) **Put** the **cursor** in the **bottom right corner** of that cell, and **move** the **mouse** until the **cursor becomes a black cross hair**.
- 3) **Click** and **drag** the **black cross hair down** the **column** until it reaches the **number two** in the **CorrectAnswer** column (rows 1-16).
- 4) **Use** the **above steps** (1-3) to **rename** the **remaining cells**, **"Right"** (rows 17-32).

The screenshots illustrate the process of editing the Target attribute in the TrialList window. The window displays a table with columns: ID, Weight, Nested, Procedure, Prime, LeftImage, RightImage, CorrectAnswer, and Target. The process involves clicking the first cell in the Target column (row 1), moving the cursor to the bottom right corner, and dragging it down to the second cell in the CorrectAnswer column (row 2). This process is repeated for all rows in the table.

ID	Weight	Nested	Procedure	Prime	LeftImage	RightImage	CorrectAnswer	Target
1	0		TrialProc	dog	dog	dog	1	Left
2	1		TrialProc	dog	dog	cat	1	Left
3	1		TrialProc	dog	dog	cow	1	Left
4	1		TrialProc	dog	dog	horse	1	Left
5	1		TrialProc	cat	cat	dog	1	Left
6	0		TrialProc	cat	cat	cat	1	Left
7	1		TrialProc	cat	cat	cow	1	Left
8	1		TrialProc	cat	cat	horse	1	Left
9	1		TrialProc	cow	cow	dog	1	Left
10	1		TrialProc	cow	cow	cat	1	Left
11	0		TrialProc	cow	cow	cow	1	Left
12	1		TrialProc	cow	cow	horse	1	Left
13	1		TrialProc	horse	horse	dog	1	Left
14	1		TrialProc	horse	horse	cat	1	Left
15	1		TrialProc	horse	horse	cow	1	Left
16	0		TrialProc	horse	horse	horse	1	Left
17	0		TrialProc	dog	dog	dog	2	Right
18	1		TrialProc	dog	cat	dog	2	Right
19	1		TrialProc	dog	cow	dog	2	Right
20	1		TrialProc	dog	horse	dog	2	Right
21	1		TrialProc	cat	dog	cat	2	Right
22	0		TrialProc	cat	cat	cat	2	Right
23	1		TrialProc	cat	cow	cat	2	Right
24	1		TrialProc	cat	horse	cat	2	Right
25	1		TrialProc	cow	dog	cow	2	Right
26	1		TrialProc	cow	cat	cow	2	Right
27	0		TrialProc	cow	cow	cow	2	Right
28	1		TrialProc	cow	horse	cow	2	Right
29	1		TrialProc	horse	dog	horse	2	Right
30	1		TrialProc	horse	cat	horse	2	Right
31	1		TrialProc	horse	cow	horse	2	Right
32	0		TrialProc	horse	horse	horse	2	Right

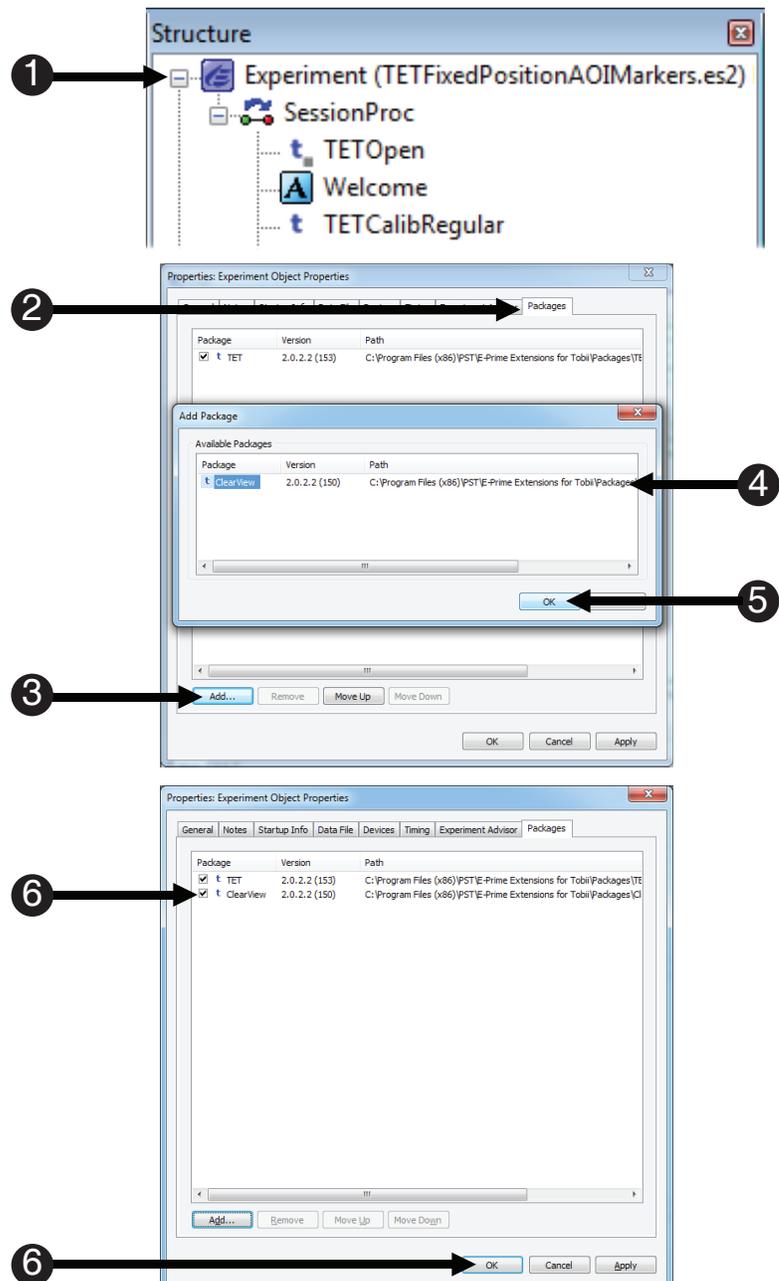
## Task 5: Add the ClearView package file to the Experiment

Open the Property Pages for the Experiment Object and use the Packages tab to add the ClearView package file to the experiment.

As was illustrated in Tutorials 1 - 4, Package Files in E-Prime are cohesive sets of E-Basic routines that are grouped together into a single file that can be maintained externally. The routines that are used to communicate with the Tobii Studio software are contained in the ClearView (CV) package file. In order to gain access to these routines, you must first add the package file to the experiment, just as you added the TET package file in Tutorial 1.

**⚠ NOTE:** As was noted in the introduction to this tutorial, this experiment is launched in parallel with a project that was created in Tobii Studio. Every project in Tobii Studio requires that calibration be performed to start the external video object. Since the E-Prime experiment already includes calls to the TET PackageFile Routine that performs calibration, you will end up performing calibration twice. If you wish to avoid the calibration step in E-Prime, highlight the TETCalibRegular PackageCall and press Delete. This will move the PackageCall to Unreferenced E-Objects, and allow the experiment to be run without first performing the extra calibration.

- 1) **Double click** the **Experiment Object** at the top of the tree in the **Structure** view.
- 2) **Click** on the **Packages tab** of the **Experiment Object** Property Pages.
- 3) **Click** the **Add...** button.
- 4) **Select** the **ClearView Package File** in the Add Package dialog.
- ⚠ NOTE:** The version numbers may differ.
- 5) **Click** the **OK** button to **dismiss** the **Add Package** dialog.
- 6) **Verify** the **ClearView package file** is listed under the **Package** column and is checked. Then **click** the **OK** button to **dismiss** the **Property Pages**.

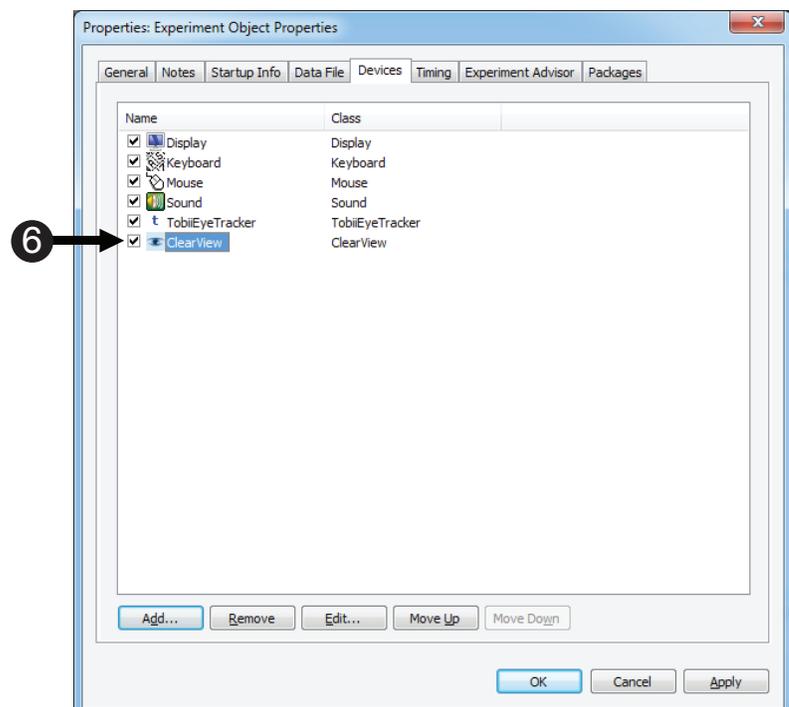
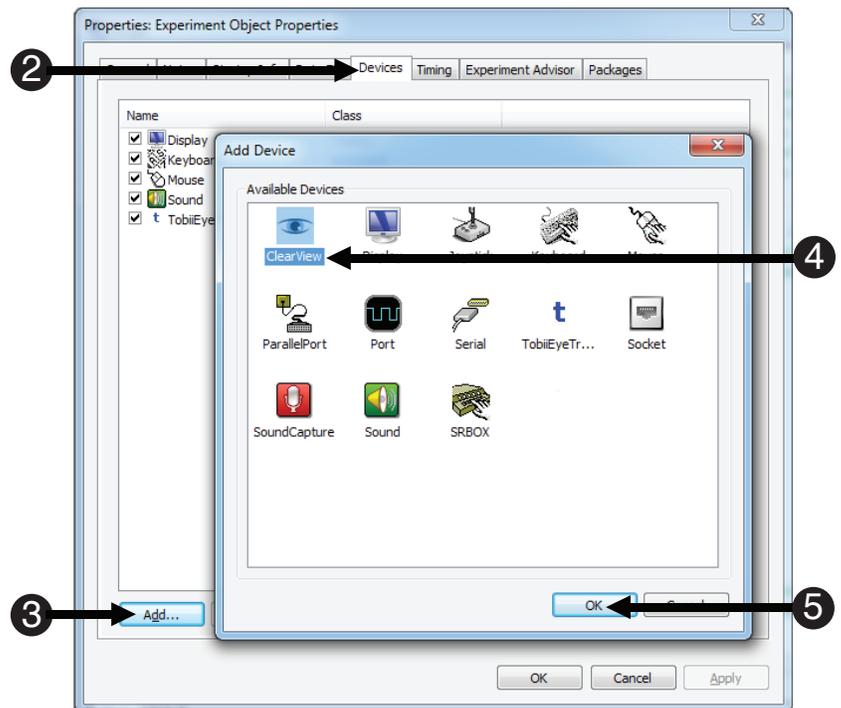


## Task 6: Add the ClearView Device to the Experiment Properties

Open the Property Pages for the Experiment Object and select the Devices tab to add the ClearView Device to the experiment.

Select the Devices Tab of the Experiment Object Property Pages to add the ClearView Device to the list of devices, and verify it is the last device shown.

- 1) **Double click** the **Experiment Object Properties** at the top of the tree in the **Structure** window.
- 2) **Click** on the **Devices** tab of the **Experiment Object Property Pages**.
- 3) **Click** the **Add...** button.
- 4) **Select** the **ClearView Device** in the **Add Device** dialog.
- 5) **Click** the **OK** button to dismiss the **Add Device** dialog.
- 6) **Verify** the **ClearView device** is listed last under the **Name** column and is checked.

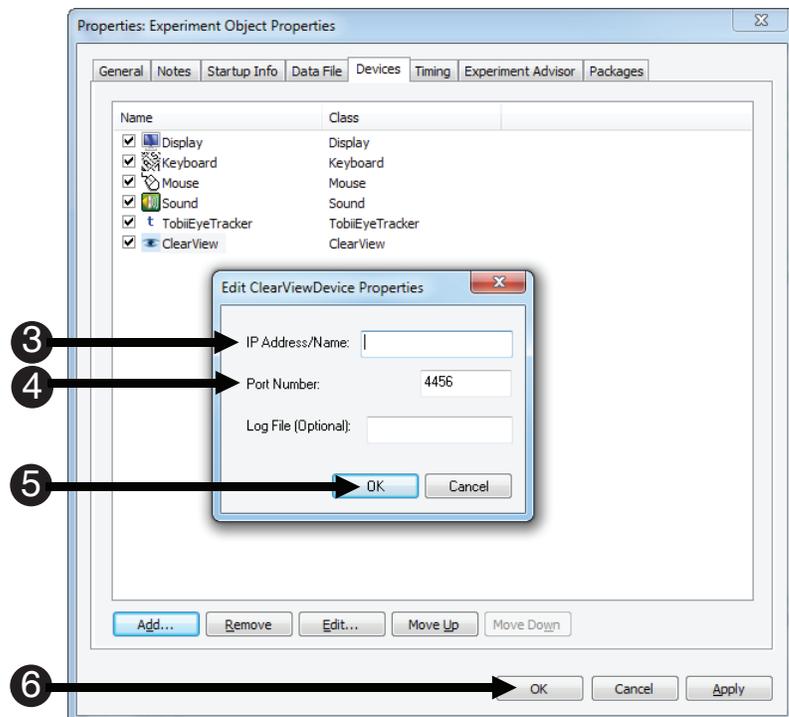
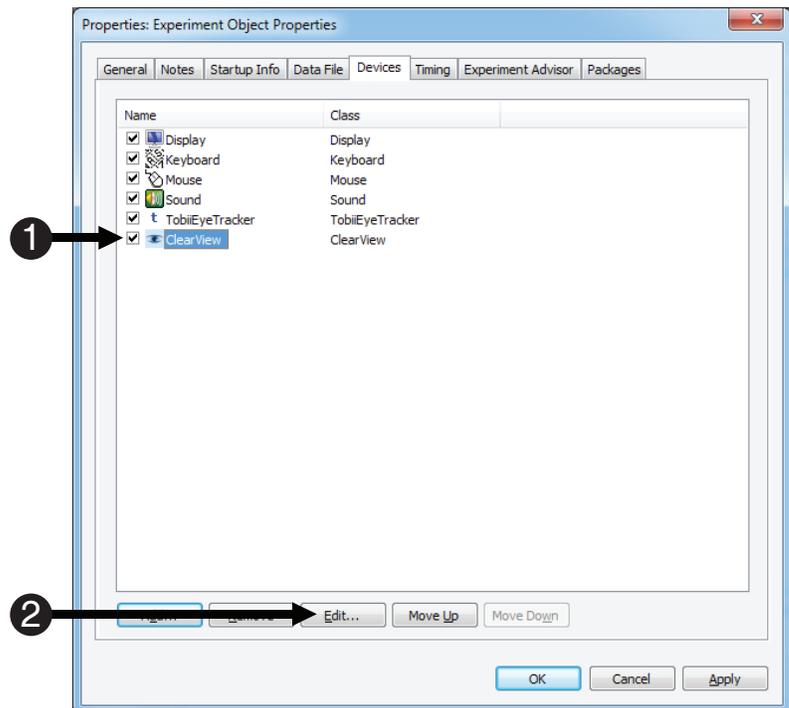


## Task 7: Edit the ClearView Device Property Pages

Open the ClearView Device Property Pages, specify the Tobii Studio machine IP Address, and confirm the TCP/IP Port number.

Now that you have added the ClearView device, you must specify the IP address of the Tobii Studio machine so that E-Prime will be able to send Scene information to Tobii Studio.

- 1) **Click** the **ClearView** device to highlight it.
- 2) **Click Edit.**
- 3) **Type** the **IP address** of the **Tobii Studio** machine (located on the back of the Eye Tracker).
- 4) **Review** the **TCP/IP Port Number**.
- 5) **Click OK** to accept changes.
- 6) **Click** the **OK** button to **dismiss** the **Add Device** dialog.

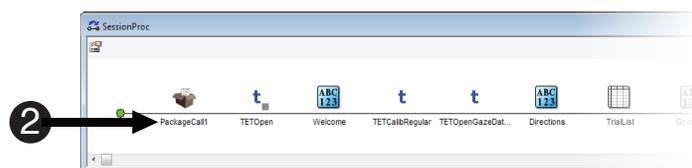
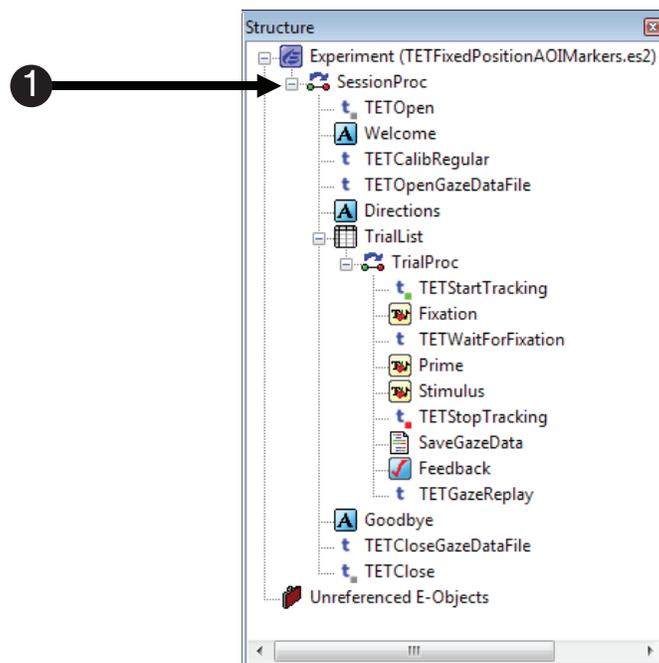


## Task 8: Add the CVInit PackageCall to initialize the ClearView Package File

Add a PackageCall at the beginning of the SessionProc to initialize the system for use with ClearView Package File. Name the object "CVInit."

The ClearView Package File contains the routines that allow E-Prime to communicate with Tobii Studio, and must be initialized at the start of the experiment by making a call to the CV\_Init routine. To call a routine in a Package File you can make the call directly using E-Basic script within an InLine object. However, the more preferable method is to drag a PackageCall from the E-Studio Toolbox, drop it at the desired location within the experiment, and edit its properties. When working with PackageCalls, we strongly recommend renaming the object to reflect the specific Package File and routine that is being referenced within the object. A common naming convention is to combine an identifier for the Package File (e.g., "CV"), with the name of the routine ("Init"). For example, the object used in this task to call the "CV\_Init" routine is named "CVInit".

- 1) **Double click** the **SessionProc** to open it in the workspace.
- 2) **Drag** a new **PackageCall** from the **E-Studio Toolbox** and **drop** it as the **first** object on the **SessionProc**, right before **TETOpen**. The object will be given a default name of **PackageCall1**.
- 3) **Click** on the **PackageCall1** to **select** it and then **press F2** to rename the object.
- 4) **Type "CVInit"**. **Press Enter** to accept the change.

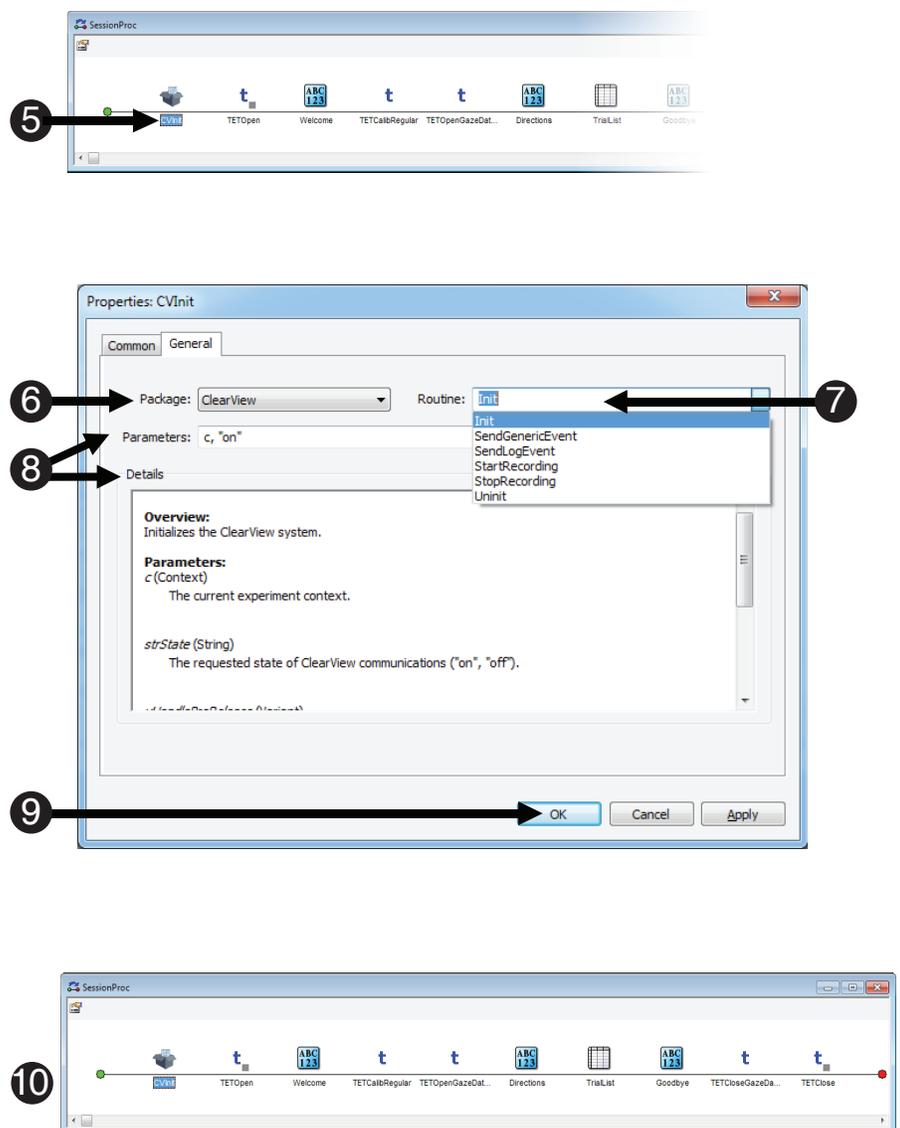


## Task 8 (continued): Add the CVInit PackageCall to initialize the ClearView Package File

Configure the CVInit PackageCall object to call the Init routine of the ClearView Package File.

The Properties dialog of the PackageCall is used to specify which Package File and Routine are to be called in each instance. After a Package is selected within the interface, the Routine dropdown list will be populated with all of the routines contained within the package. After you select a Routine, the Parameters field will be set to the default parameters and the Details field will be filled with the text that the Package File author included for the selected routine. You can refer to the Details field for information about each parameter in the list (any parameter in double quotes indicates string data). The CVInit PackageCall includes a parameter that allows you to turn communications with Tobii Studio on/off. The default is on.

- 5) **Double click** the CVInit PackageCall to **display** its Property Pages.
- 6) **Select ClearView** from the Package dropdown list.
- 7) **Select Init** from the Routine dropdown list.
- 8) **Review** the CVInit parameters listed in the Parameters and Details fields.
- 9) **Click** the OK button to **accept** the changes and **dismiss** the Property Pages.
- 10) **Confirm** your SessionProc is identical to the example shown.



## Task 9: Add the CVStartRecording PackageCall

Add a *PackageCall* before the *Welcome Object* to initialize the system for use with *ClearView Package File*. Name the object *CVStartRecording*.

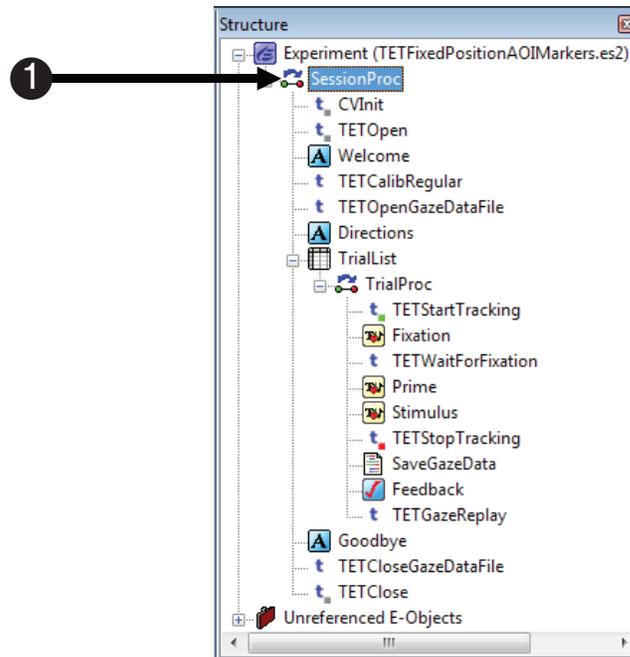
The *CVStartRecording PackageCall* tells the external video object in Tobii Studio to record the video signal that is sent from E-Prime to the capture card. This *PackageCall* should be placed at the location in the experiment where it is appropriate to begin recording data. In this experiment, the appropriate location is immediately before the *Directions Text Object*.

1) If the **SessionProc** is not already open, **double click** the **SessionProc** to **open** it in the workspace.

2) **Drag** a new **PackageCall** from the **E-Studio Toolbox** and **drop** it immediately **before** the **Directions object**. The **Object** will be given a default name of **PackageCall1**.

**⚠ NOTE:** In order to include the *Directions* in the recording, the *CVStartRecording PackageCall* needs to be placed before the *Direction* object.

3) **Click** on the **PackageCall1** to **select** it and then **press F2** to **rename** the object. **Type "CVStartRecording"** and then **press Enter** to **accept** the change.

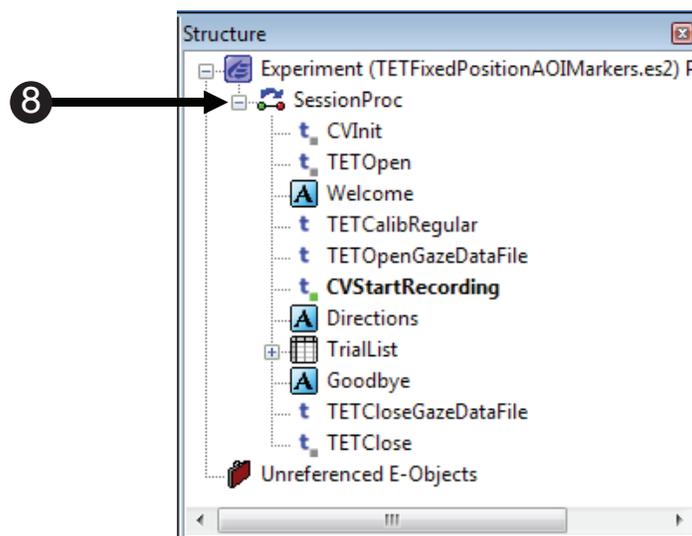
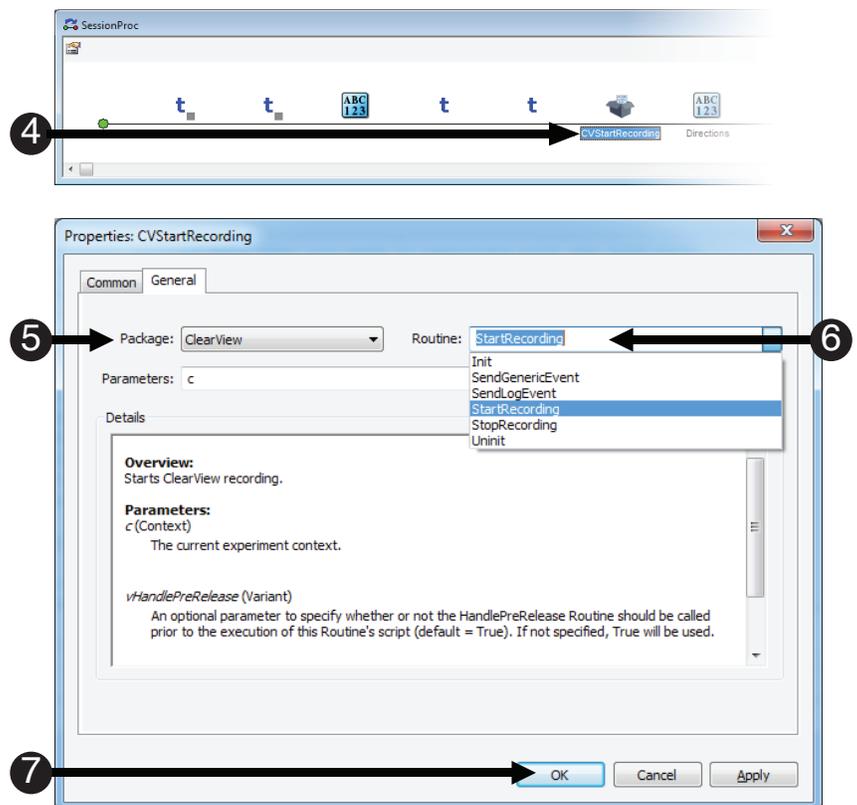


## Task 9 (continued): Add the CVStartRecording PackageCall

Configure the CVStartRecording PackageCall to the StartRecording routine of the ClearView Package File and accept the default Parameters list.

The CVStartRecording PackageCall only has one parameter, c. This parameter sets the current experiment context. Accept the default parameter and close the Property Pages dialog.

- 4) **Double click** the **CVStartRecording PackageCall** to **display** its **Property Pages**.
- 5) **Select ClearView** from the **Package** dropdown list.
- 6) **Select StartRecording** from the **Routine** dropdown list.
- 7) **Click the OK** button to **accept** the **changes** and **dismiss** the **Property Pages**.
- 8) **Confirm** your **SessionProc** is **identical** to the **example** shown here.

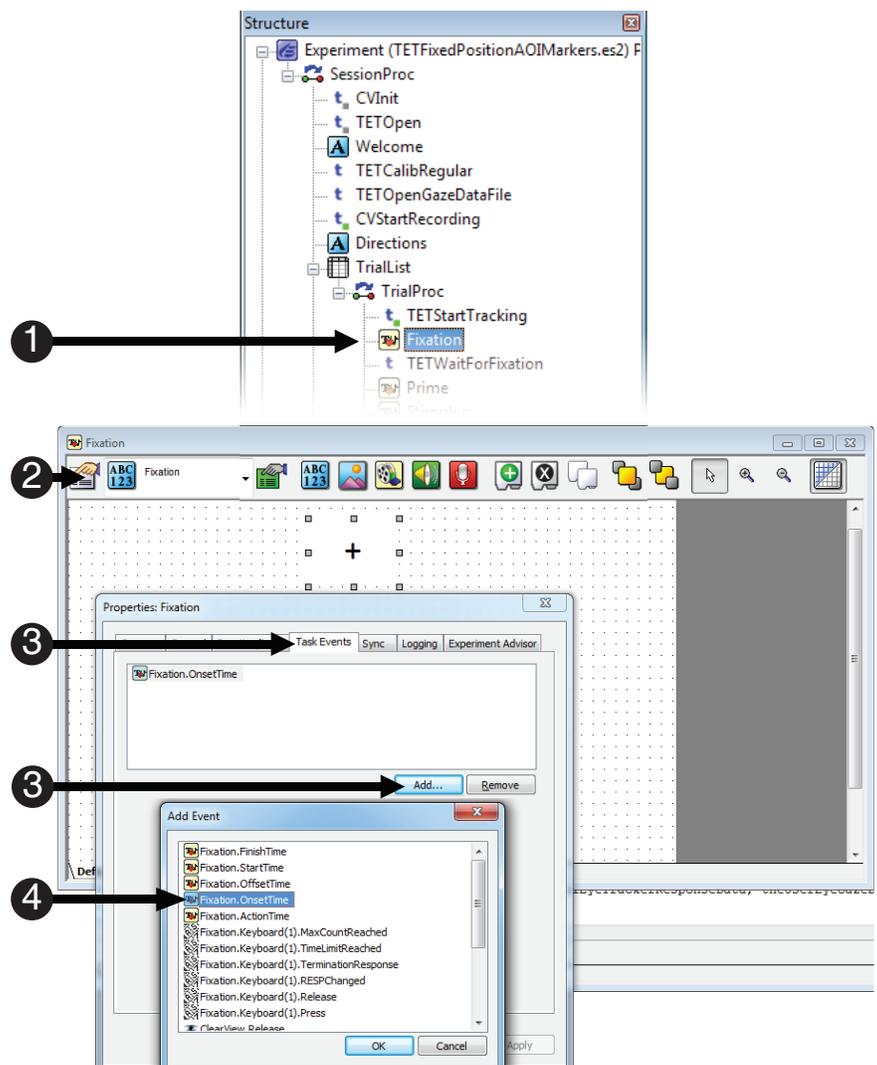


## Task 10: Add a Task Events to the Fixation object

Define the "Event" component of the Task Event by specifying the `Fixation.OnsetTime` as the triggering event.

As described in the introduction to this section, this tutorial uses E-Prime 3.0's Task Events feature to send Scene markers to Tobii Studio. Tobii Studio requires both a Scene Start and Scene Stop marker in order to perform Scene-based analyses of gaze data. Therefore, each Scene requires two related Task Events, one for Scene Start and another for Scene Stop. The first scene to be defined is the Fixation Scene. The Fixation Object's Onset Time, or the time when the Fixation object begins its action of being displayed, is used as the triggering event to mark the Scene Start. (As you will see later, this same event is used to mark the start of other scenes as well.) The trigger event to mark the stop of the Fixation Scene is the Prime Object's Onset Time. Using the Prime Object's Onset Time, rather than the Fixation Object's Offset Time, provides a more precise time stamp of when the Fixation Scene ends, because the Prime Object overwrites the information that is currently being displayed with the Fixation Object. The offset time for Fixation does not indicate when the Fixation object is no longer visible; it identifies when the Fixation Object is no longer executing. For more details about when information on the screen is updated, see **Section 4.5.3: Stimulus Presentation Solutions** in the *E-Prime 3.0 User's Guide*. Note that the `CVSendLogEvent PackageCall` could also have been used to send Scene marker information to Tobii Studio. However, we recommend using the Task Events method because it intrinsically accounts for issues that would otherwise need to be handled with the coordination of `PackageCall` placement and other timing parameters.

- 1) **Double click** the **Fixation object** to open it in the workspace.
- 2) **Click** the white **Property Pages** button.
- 3) **Select** the **Task Events** tab, and then **click** the **Add...** button.
- 4) **Double click** `Fixation.OnsetTime`.

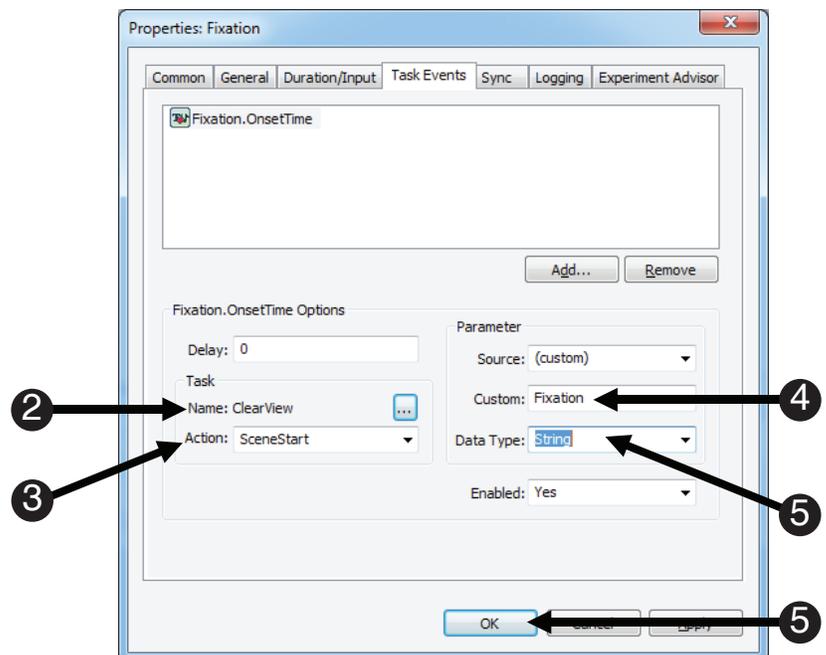
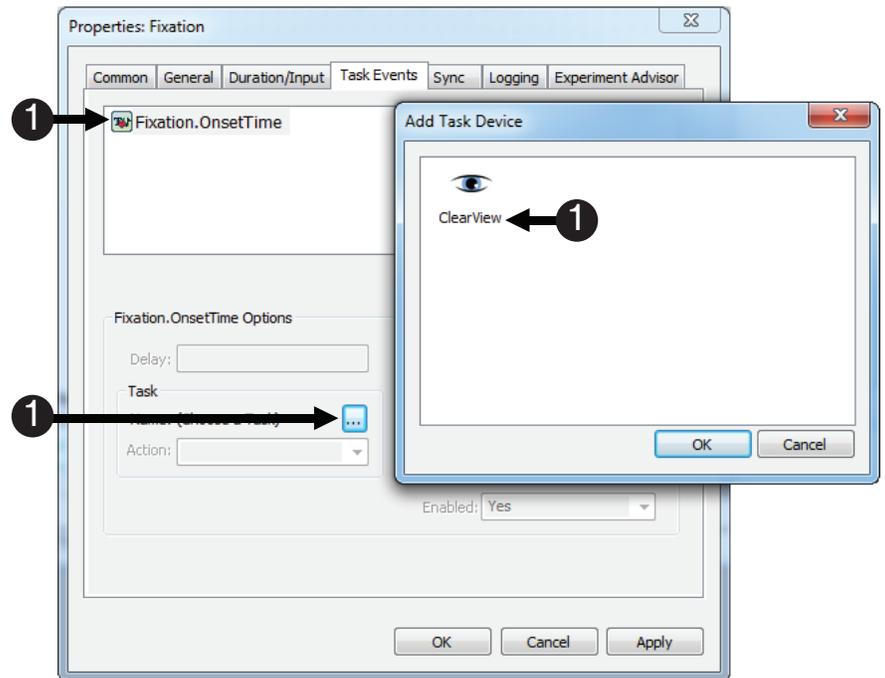


## Task 11: Edit the Fixation.StartOnsetTime Task Event

Define the "Task" component of the Task Event by specifying the Fixation SceneStart as the information to be sent to Tobii Studio.

The next step is to configure the Task Event to send the Scene Start information to Tobii Studio. In this case, we want to send a string that identifies the start of the Fixation Scene when the Fixation object begins to display.

- 1) **Select** the **Fixation.OnsetTime** task event. **Click** the ... button next to the **Name** field. **Double click** the **ClearView** icon.
- 2) **Note** the **Name** field **reads ClearView**, indicating that the **ClearView** device will **receive** the **Task** information.
- 3) **Select SceneStart** from the **Action** field.
- 4) **Edit** the **Custom** field to **read Fixation**.
- 5) **Select String** as the **Data** type, and **click OK**.

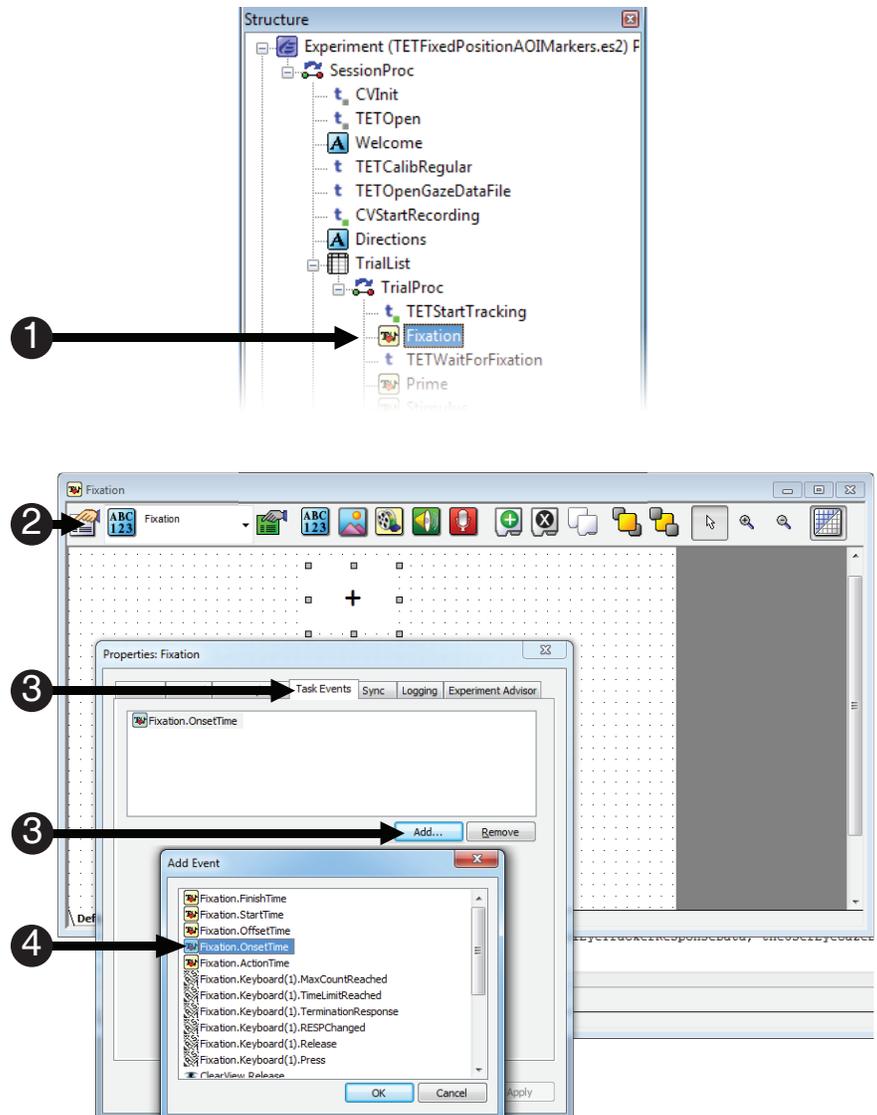


## Task 12: Add another Task Event to the Fixation object

Define another Task Event based on the Fixation Object's OnsetTime; this will mark the start of the Trial Scene.

A second Task Event, which also uses the Fixation.OnsetTime as the triggering event, is added to the Fixation Object. This second Task Event is used to identify the start of the Trial Scene. Here we create a trial scene for each target type ("Left" or "Right"), using the Target Attribute that was created in Task 3 of this tutorial. Therefore, we have two scenes that start when the Fixation object beings to display. These scenes will end, however, with different events on different objects, as you will see shortly.

- 1) **Double click** the **Fixation object** to open it in the workspace.
- 2) **Click** the white **Property Pages** button.
- 3) **Select** the **Task Events** tab, and then **click** the **Add...** button.
- 4) **Double click** Fixation.OnsetTime.

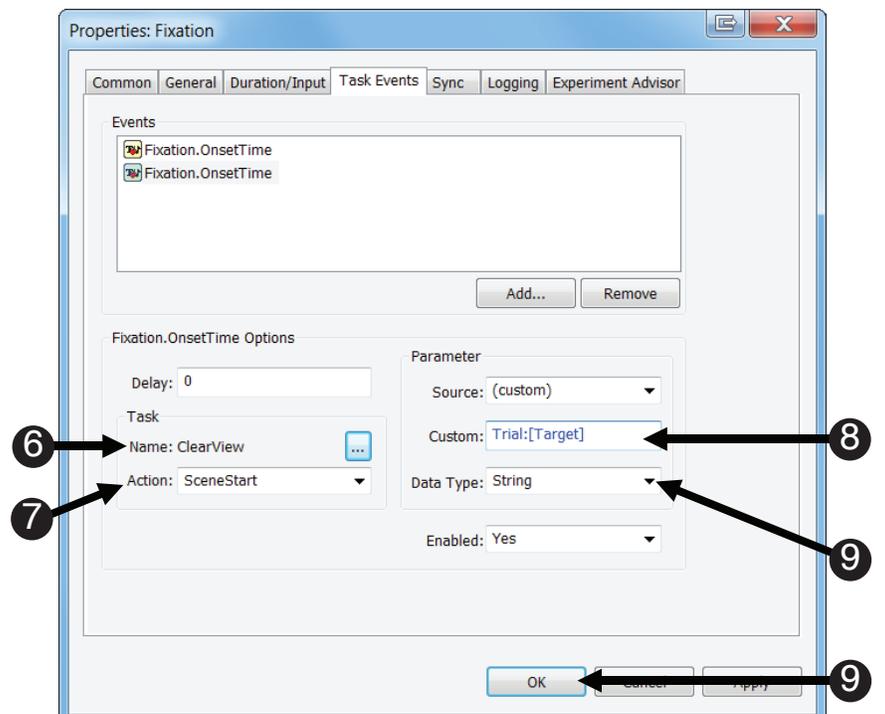
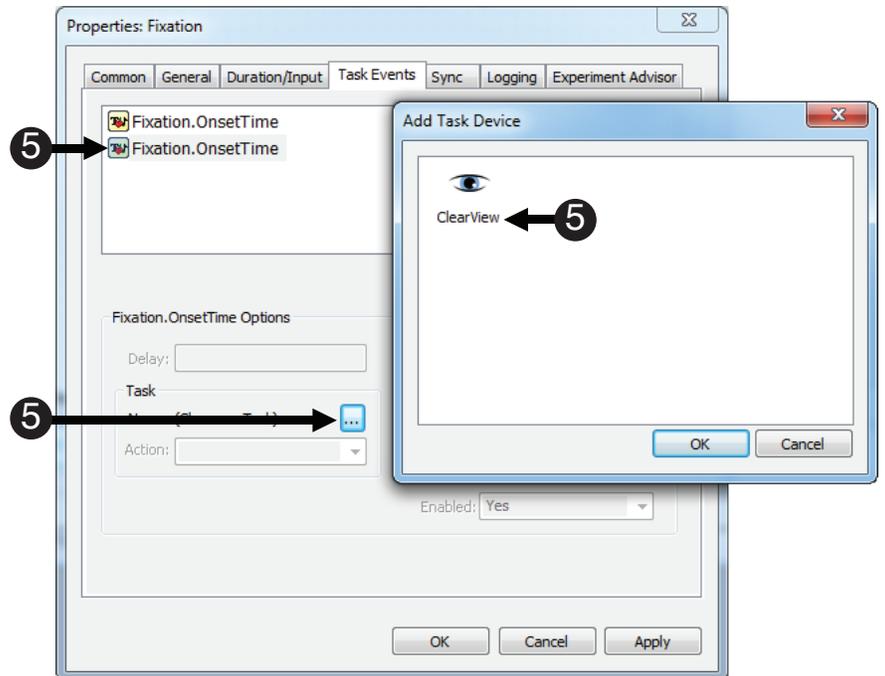


## Task 12 (continued): Add another Task Event to the Fixation

Define another Task Event based on the Fixation Object's OnsetTime; this will mark the start of the Trial scene.

The Target attribute becomes part of the Scene marker that is sent to Tobii Studio. When the Target attribute equals "Left", then the Scene marker will equal "Trial:Left"; when the Target attribute equals "Right", then the Scene marker will equal "Trial:Right".

- 5) **Select** the second **Fixation.OnsetTime** task event. **Click** the ... button next to the **Name** field. **Double click** the **ClearView** icon to **select ClearView** as the **Task device**.
- 6) **Note** the **Name** field **reads ClearView**.
- 7) **Select SceneStart** from the **Action** field.
- 8) **Edit** the **Custom** file to **read: Trial:[Target]**.
- 9) **Select String** as the **Data type**, and **click OK**.

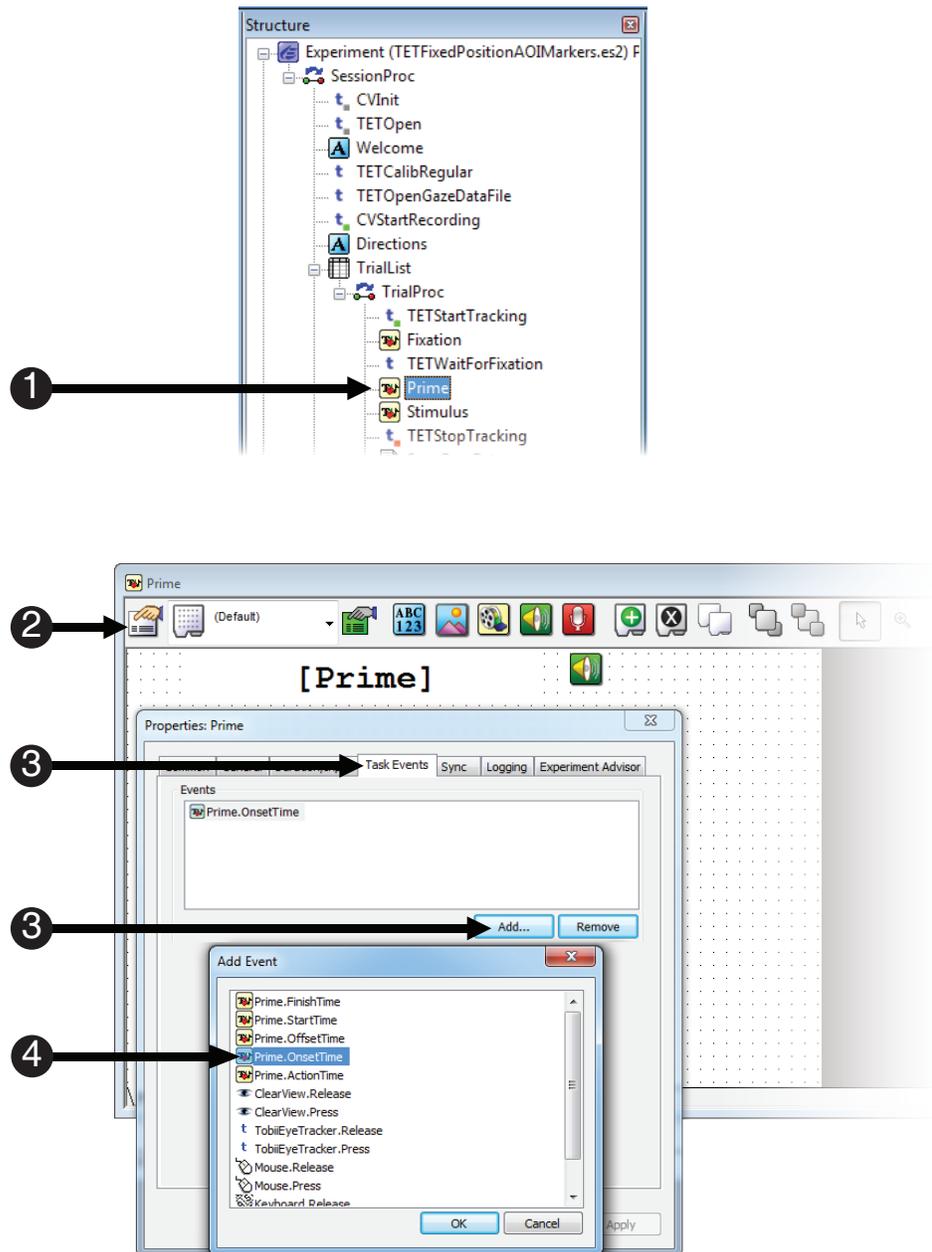


## Task 13: Add a Fixation SceneStop Task Event to the Prime Object

Define the "Event" component of the Task Event by specifying the Prime.OnsetTime as the triggering event.

Here we define the end of the Fixation scene. The Prime Object's OnsetTime is the triggering event to mark the end of the Fixation scene that was started in Task 11 of this tutorial.

- 1) **Double click** the **Prime Object** to **open** it in the workspace.
- 2) **Click** the white **Property Pages** button.
- 3) **Select** the **Task Events** tab, and then **click** the **Add...** button.
- 4) **Double click** **Prime.OnsetTime**.

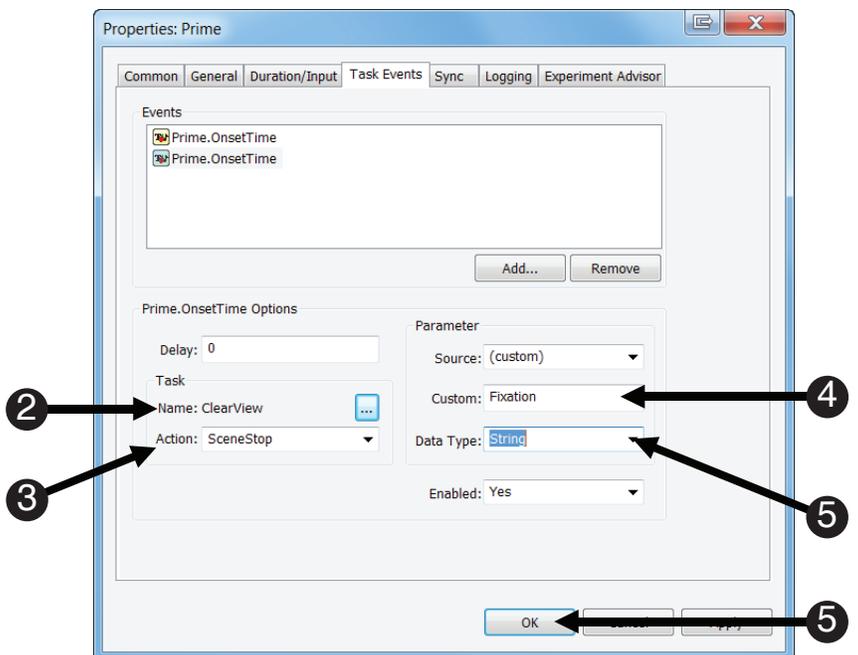
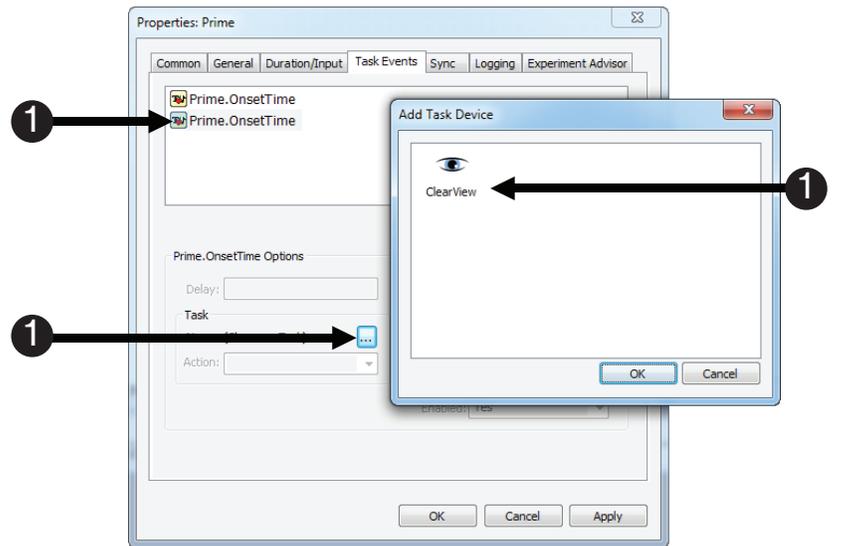


## Task 14: Edit the Fixation SceneStop Task Event

Define the "Task" component of the PrimeObject Task Event by specifying the Fixation SceneStop marker as the information to be sent to Tobii Studio.

This Task Event defines the end of the Fixation scene. Therefore, we send the SceneStop marker for the Fixation scene to Tobii Studio when the Prime Object begins to execute.

- 1) **Select** the second **Prime.OnsetTime** task event. **Click** the ... button next to the **Name** field. **Double click** the **ClearView** icon.
- 2) **Note** the **Name** field **reads** **ClearView**.
- 3) **Select** **SceneStop** from the **Action** field.
- 4) **Edit** the **Custom** field to **read** **Fixation**.
- 5) **Select** **String** as the **Data** type, and **click** **OK**.

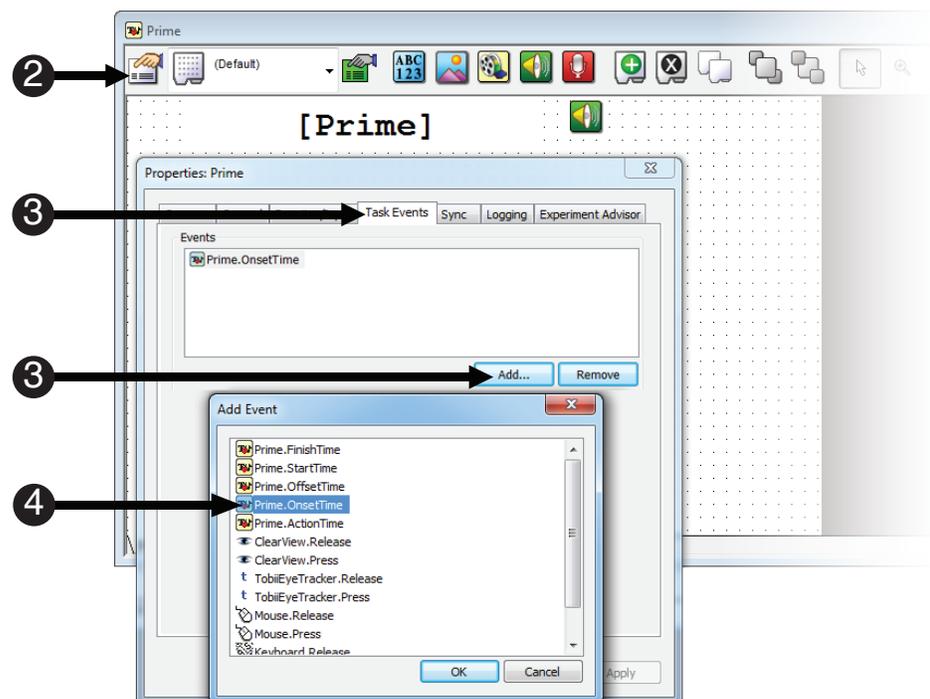
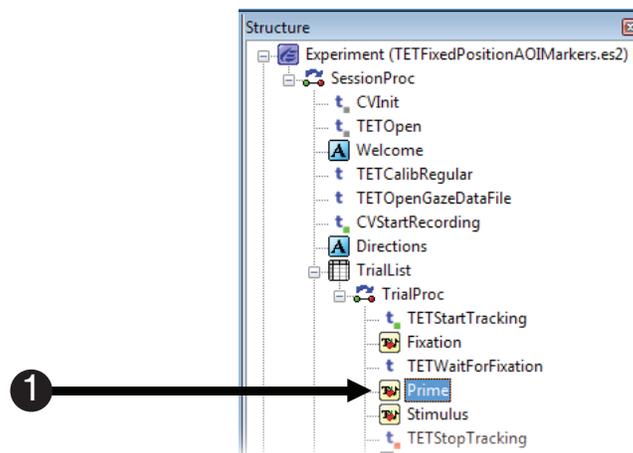


## Task 15: Create the Prime Scene by adding another Task Event to the Prime Object

Define the "Event" component of a new Task Event for the Prime Object by specifying the *Prime.OnsetTime* as the triggering event for the Prime scene.

This step will add another .OnsetTime task event to the Prime Object. This task event will send the onset time of the Prime Object from E-Prime to Tobii Studio. This Task Event sends the SceneStart marker for the Prime scene when the Prime Object begins to execute. As you will see later, the SceneStop marker for the Prime scene will be sent when the Stimulus Object begins to execute.

- 1) If necessary, **double click** the **Prime** to **open** it in the workspace.
- 2) **Click** the white **Property Pages** button.
- 3) **Select** the **Task Events** tab, and then **click** the **Add...** button.
- 4) **Double click** **Prime.OnsetTime**.

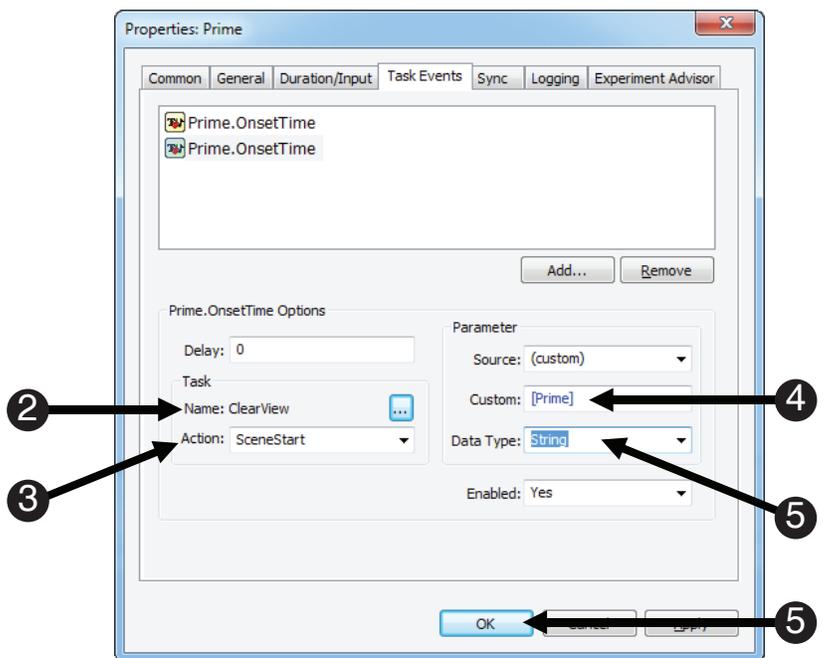
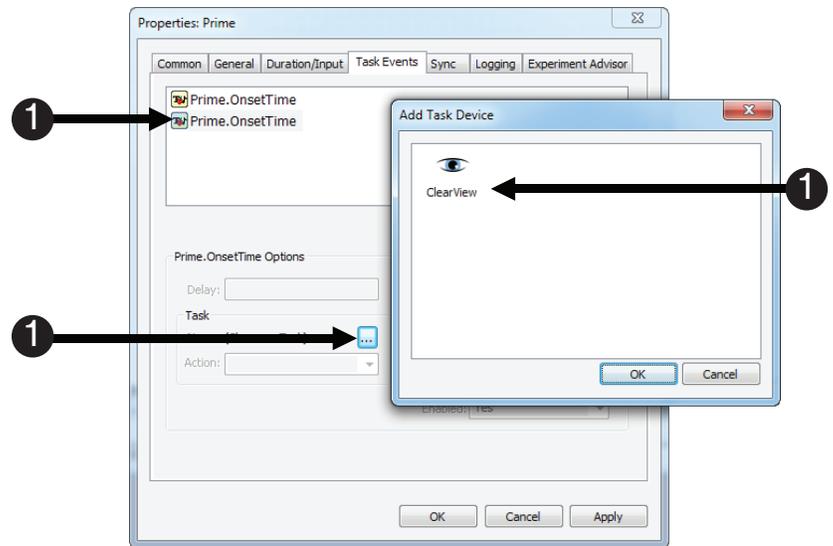


## Task 16: Edit the Prime Scene Task Event

Define the "Task" component of the Task Event by specifying the Prime SceneStart marker as the information to be sent to Tobii Studio.

The next step is to edit the Prime.OnsetTime Task Event so E-Prime knows what information to send to Tobii Studio. In this case we want to send an Attribute identifying the Prime, so we will edit the Task Event parameters to do so. Recall that the Prime attribute is assigned to the Prime word that appears on each trial. Therefore, sending the Prime attribute will result in scenes that are labeled "dog", "cat", "cow", or "horse".

- 1) **Select** the second **Prime.OnsetTime** task event. **Click** the ... button next to the **Name** field. **Double click** the **ClearView** icon.
- 2) **Note** the **Name** field **reads ClearView**.
- 3) **Select SceneStart** from the **Action** field.
- 4) **Edit** the **Custom** field to **read [Prime]**.
- 5) **Select String** as the **Data** type, and **click OK**.

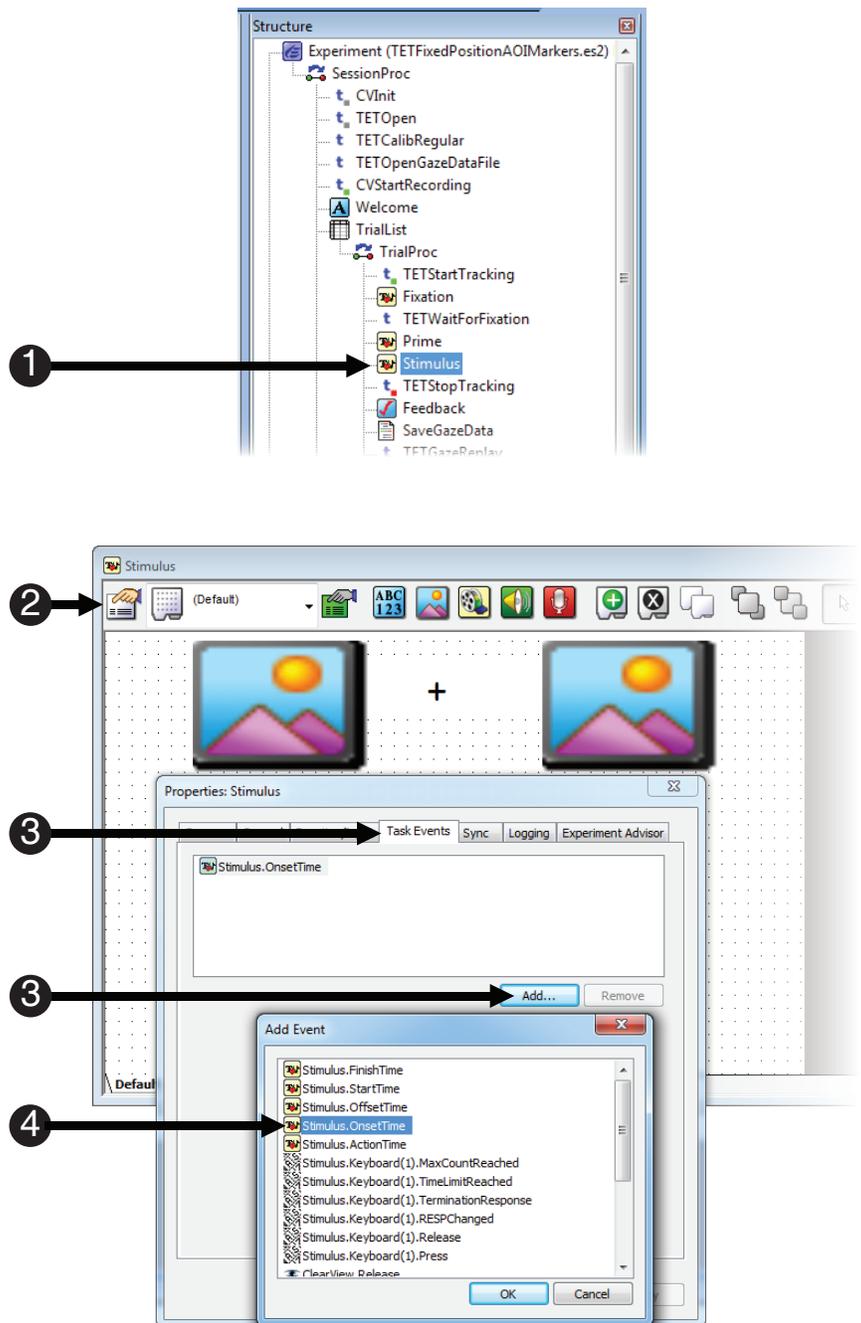


# Task 17: Add the Prime SceneStop Task Event to the Stimulus Object

Define the "Event" component of a new Task Event by specifying the Stimulus.OnsetTime the triggering event.

This Task Event sends the Prime SceneStop signal at the onset time of the Stimulus Object to Tobii Studio. This ensures that the Prime scene in Tobii Studio has a starting and ending point. Like the last SceneStop that we sent for the Fixation scene, the SceneStop is triggered during the onset of the next display object in the trial procedure (Stimulus) instead of the offset of the object, which marked the start of the scene (Prime).

- 1) **Double click** the **Stimulus** to **open** it in the workspace.
- 2) **Click** the white **Property Pages** button.
- 3) **Select** the **Task Events** tab, and then **click** the **Add...** button.
- 4) **Double click** **Stimulus.OnsetTime**.

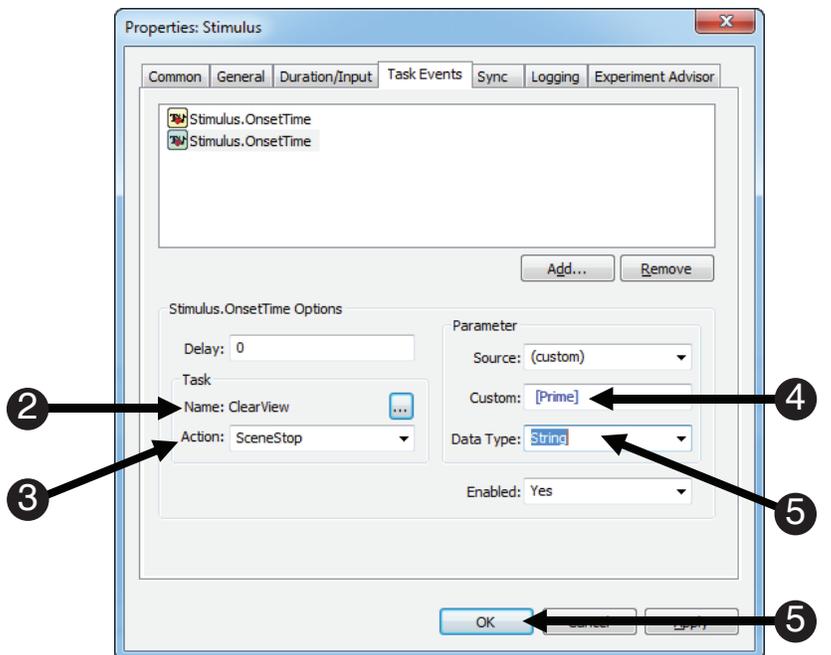
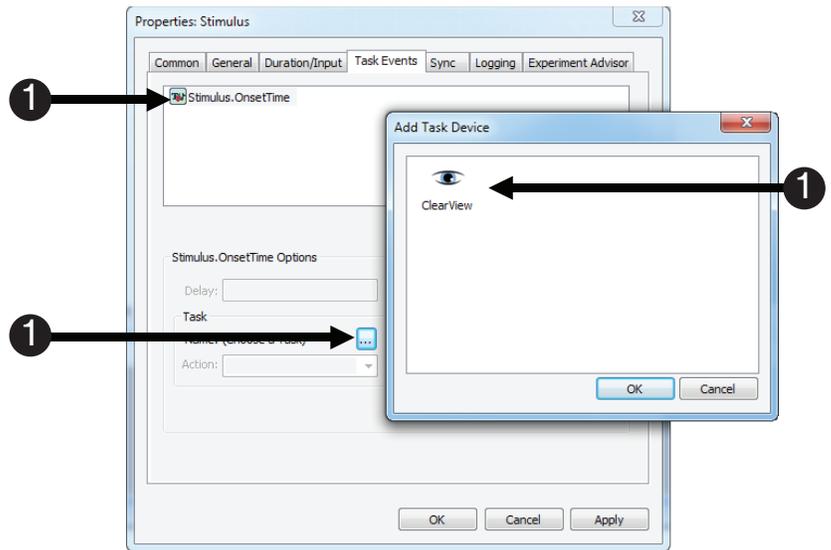


## Task 18: Edit the Stimulus.OnsetTime Task Event

Define the "Task" component of the Task Event by specifying the Prime SceneStop marker as the information to be sent to Tobii Studio.

This Task Event defines the end of the Prime scene. Therefore, we send the value of the [Prime] attribute to Tobii Studio. As was done earlier with the Fixation scene, sending both a SceneStart and SceneStop marker is required in order for Tobii Studio to support a scene-based analysis.

- 1) **Select** the **Stimulus.OnsetTime** task event. **Click** the ... button next to the **Name** field. **Double click** the **ClearView** icon.
- 2) **Note** the **Name** field **reads** **ClearView**.
- 3) **Select** **SceneStop** from the **Action** field.
- 4) **Edit** the **Custom** field to **read** **[Prime]**.
- 5) **Select** **String** as the **Data** type, and **click** **OK**.

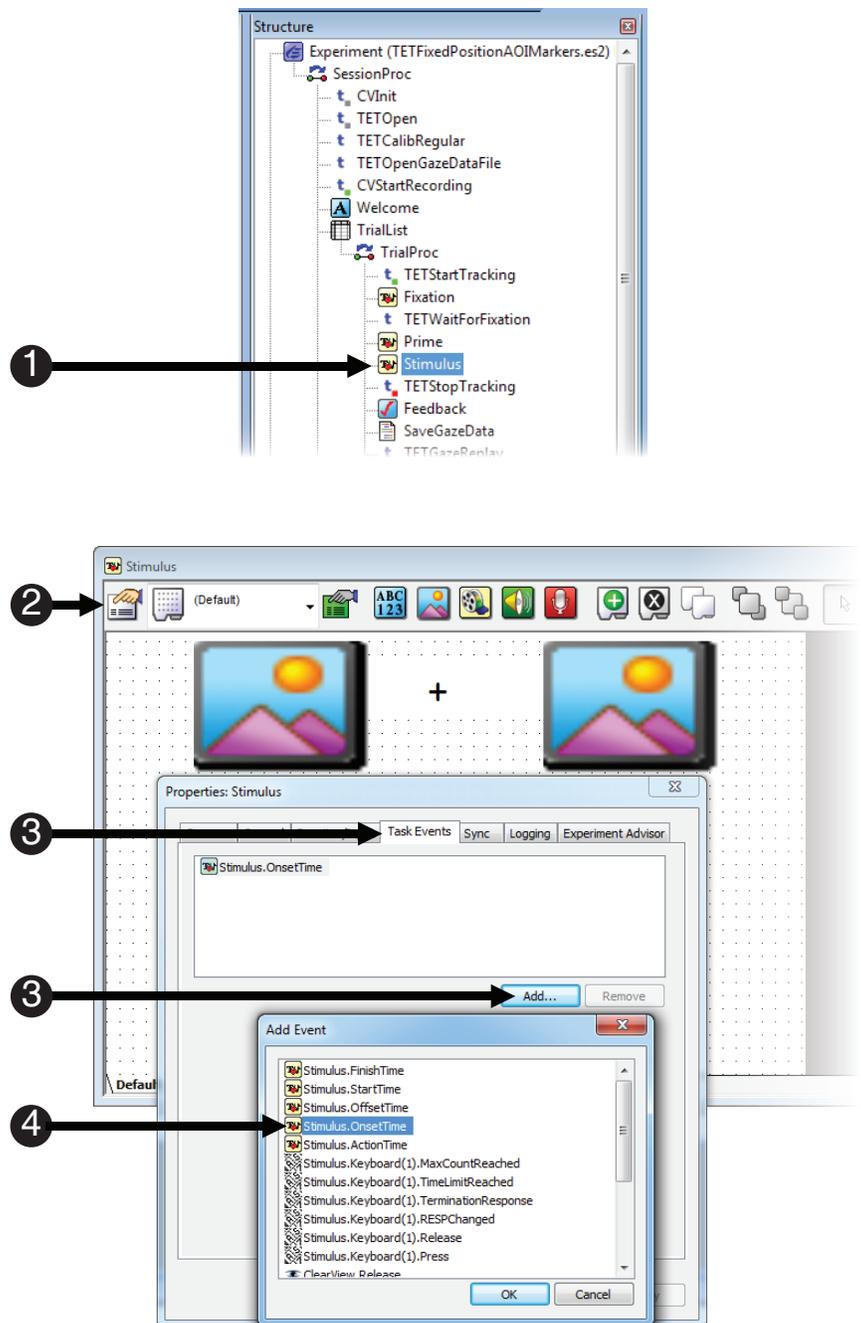


## Task 19: Create multiple new SceneStarts with one Task Event on the Stimulus Object

Define the "Event" component of a new Task Event that will be used to create multiple scenes.

Unlike the last few Task Events that have been created, the Task Event created below utilizes a special feature of E-Prime Extensions for Tobii, which enables the start of multiple scenes to occur within a single Task Event.

- 1) If necessary, **double click** the **Stimulus** to **open** it in the workspace.
- 2) **Click** the white **Property Pages** button.
- 3) **Select** the **Task Events** tab, and then **click** the **Add...** button.
- 4) **Double click** **Stimulus.OnsetTime**.

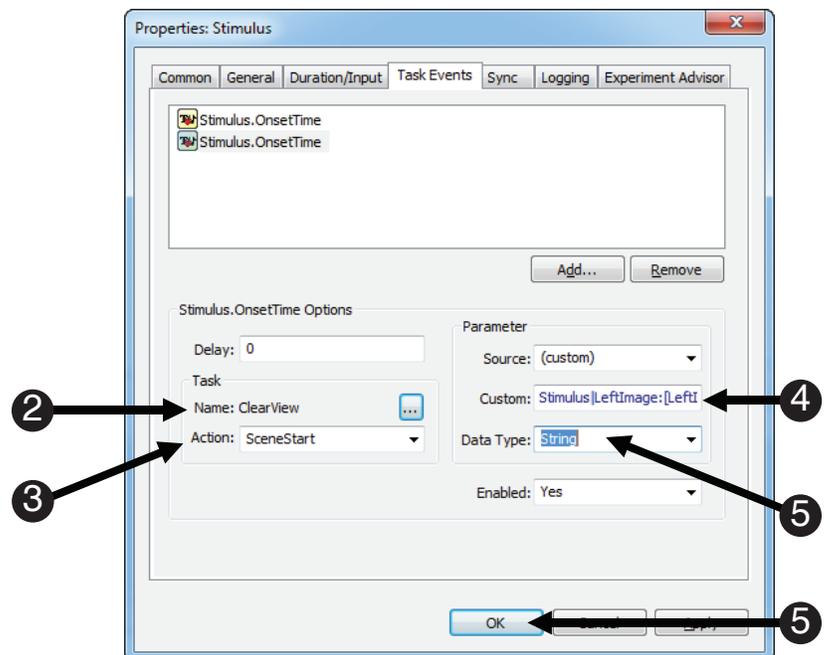
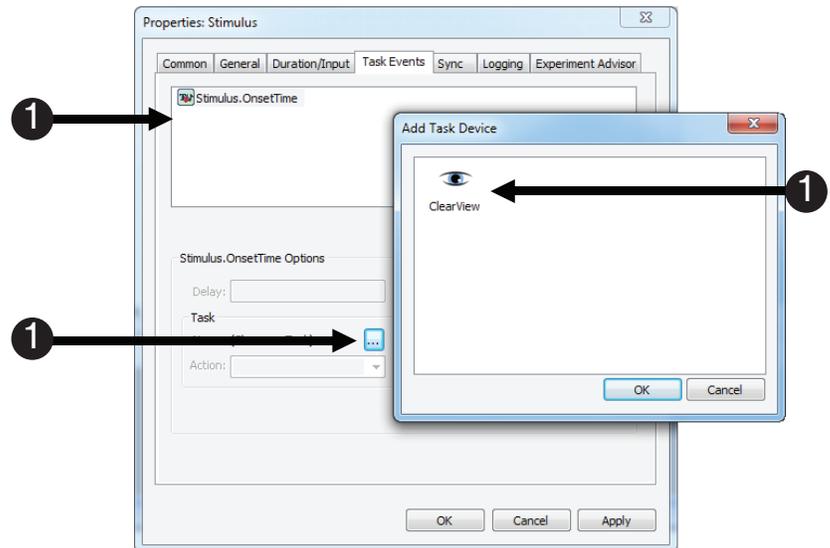


## Task 20: Edit the Stimulus.OnsetTime Task Event

Define the "Task" component of the event as sending multiple SceneStart markers to Tobii Studio.

The next step is to define the Task component for the Stimulus Object Event that was created in the previous task. In this case, we want to perform multiple Tasks, i.e. send multiple SceneStart markers, for a single Event. A special feature of E-Prime Extensions for Tobii is the use of the pipe character (|) as a delimiter to specify multiple values in the Custom field. Separating the names of each Scene marker enables E-Prime to send multiple SceneStart triggers to Tobii Studio with only one Task Event. The first scene marker to be sent is the Stimulus Scene. This will allow you to conduct scene-based analyses on the entire Stimulus Slide object. The next two scenes describe the right and left images, allowing scenes with similar stimuli to be analyzed together. Finally, we create a scene called RESP that focuses on the period of time just before the participant makes a response.

- 1) **Select** the second **Stimulus.OnsetTime** task event. **Click** the ... button next to the **Name** field. **Double click** the **ClearView** icon.
- 2) **Note** the **Name** field **reads** **ClearView**.
- 3) **Select** **SceneStart** from the **Action** field.
- 4) **Edit** the **Custom** field to **read** **Stimulus|LeftImage:[LeftImage]|RightImage:[RightImage]|RESP**.
- 5) **Select** **String** as the **Data** type, and **click** **OK**.

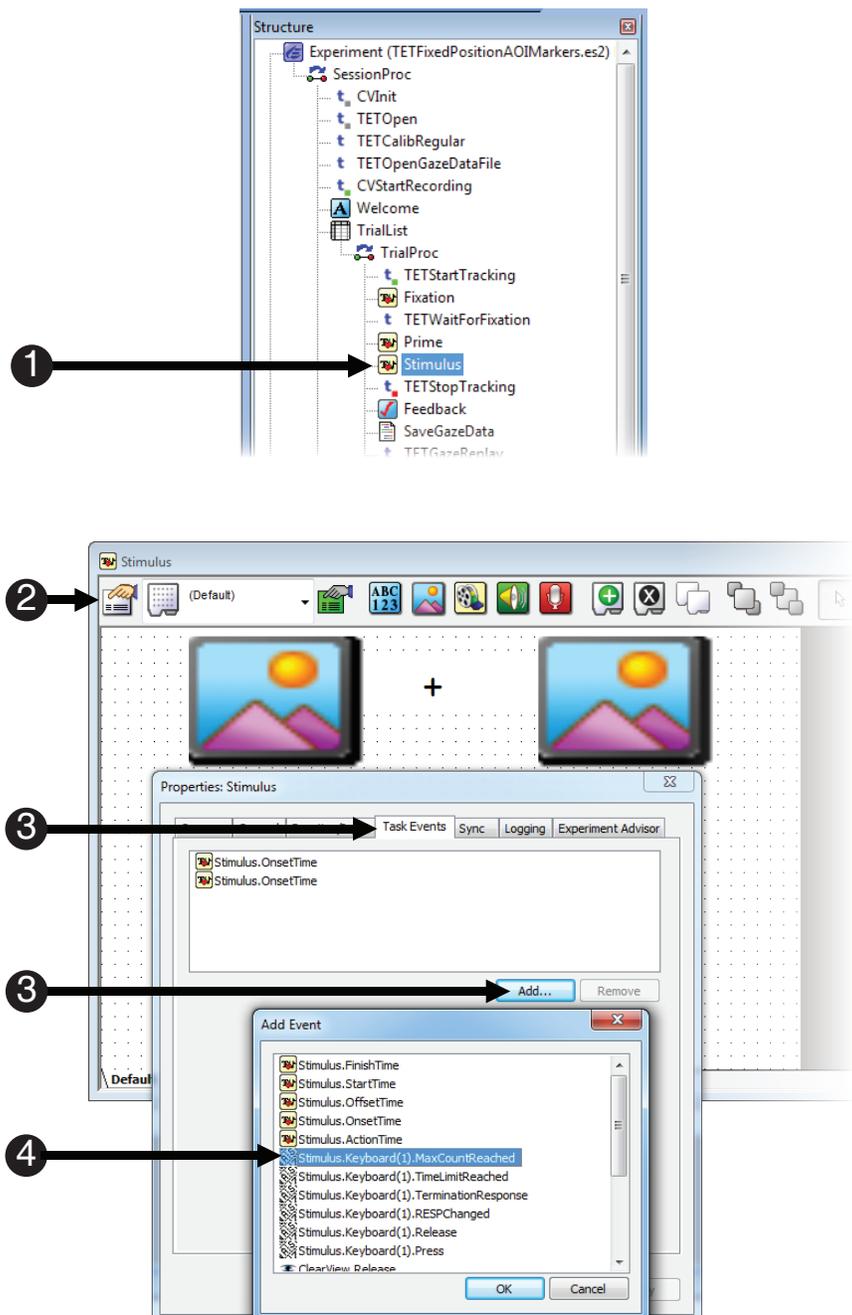


## Task 21: Add another Task Event to the Stimulus Object

Define the "Event" component as the `Stimulus.Keyboard(1).MaxCountReached`; this event will trigger several important data-related scenes.

The purpose of this Task Event is to define the end of the RESP scene in Tobii Studio. We want to end the RESP scene when the Input Mask is terminated for the Stimulus Object. This can occur in one of two ways – either a response is made, or the object's display duration is reached. This Task Event identifies the former, when the maximum number of responses has been made to the Stimulus Object. (In this case, the maximum number of responses that can be made to the Stimulus is one; e.g., a single keypress terminates the InputMask). This event is triggered with `Stimulus.Keyboard(1).MaxCountReached`. The trigger that comes across when the max count is reached will accurately reflect when a response was made.

- 1) If necessary, **double click** the **Stimulus** to **open** it in the workspace.
- 2) **Click** the white **Property Pages** button.
- 3) **Select** the **Task Events** tab, and then **click** the **Add...** button.
- 4) **Double click** `Stimulus.Keyboard(1).MaxCountReached`.

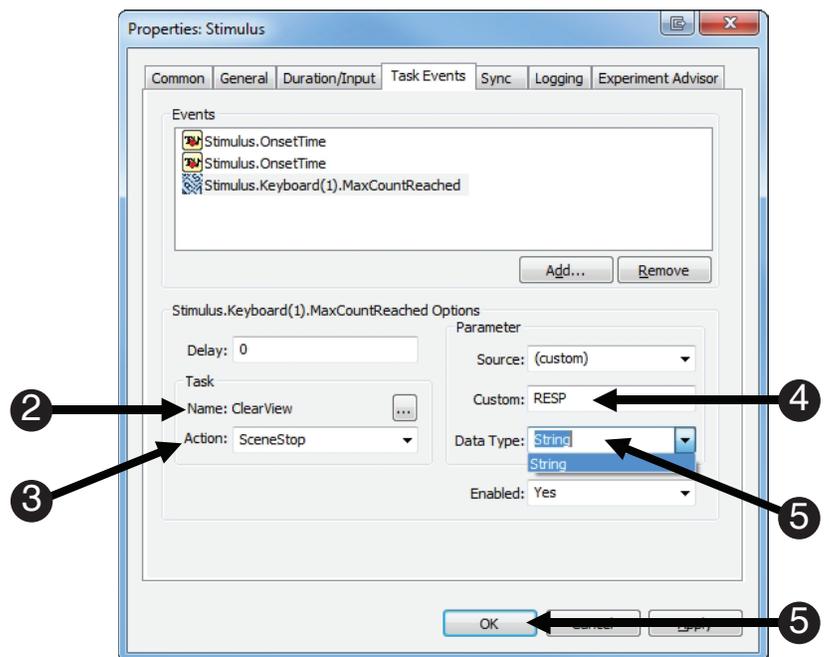
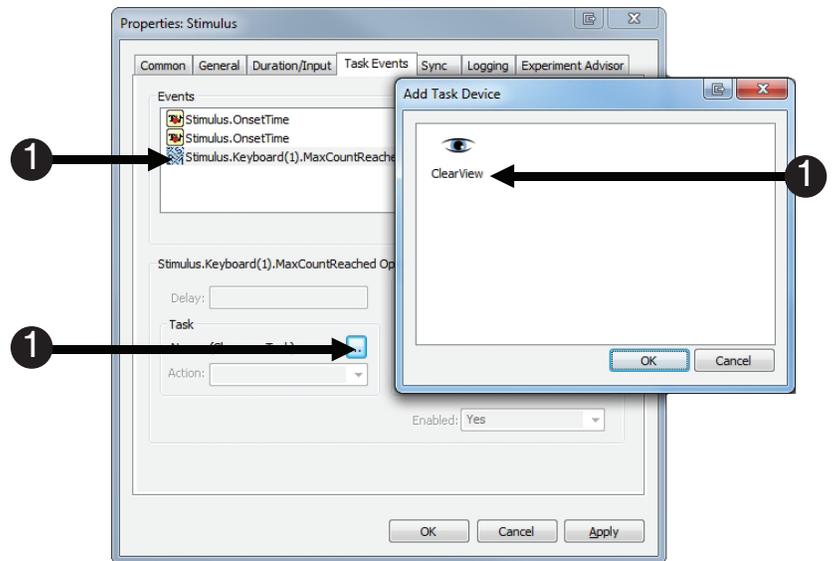


## Task 22: Edit the Task associated with the Stimulus.Keyboard(1).MaxCountReached Event

Define the "Task" component of the Task Event as sending the RESP SceneStop marker to Tobii Studio.

This Task Event will ensure that the RESP scene has a SceneStop marker when the participant responds. We will also add a Task Event to the experiment that handles the instances when a participant does not respond in the next two Tutorial Tasks.

- 1) **Select** the **Stimulus.Keyboard(1).MaxCountReached** task event. **Click** the ... button next to the **Name** field. **Double click** the **ClearView** icon.
- 2) **Note** the **Name** field **reads** **ClearView**.
- 3) **Select** **SceneStop** from the **Action** field.
- 4) **Edit** the **Custom** field to **read** **RESP**.
- 5) **Select** **String** as the **Data** type, and **click** **OK**.

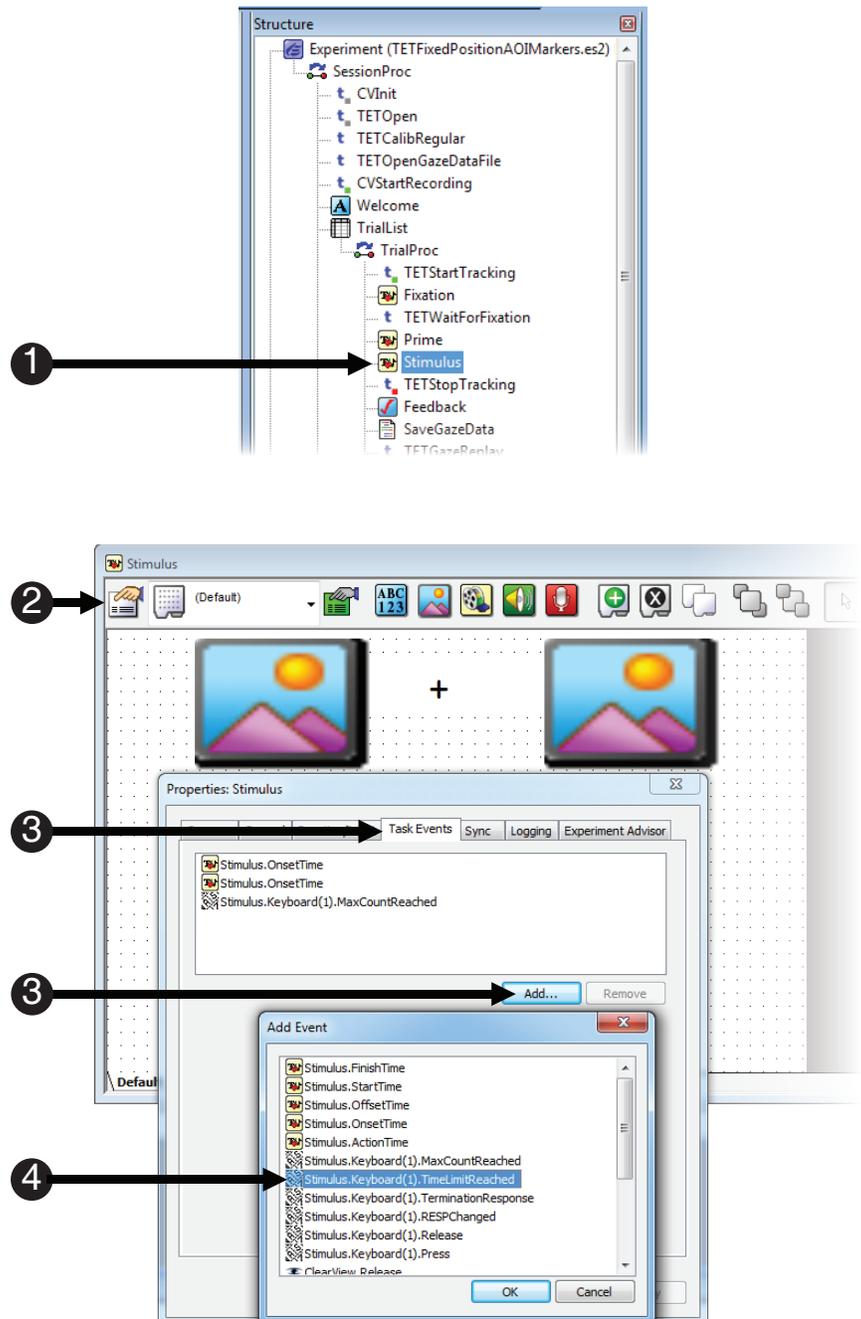


## Task 23: Add another Task Event to the Stimulus Object

Define the "Event" component as `Stimulus.Keyboard(1).TimeLimitReached`; this occurs when no response has been made.

This Task Event will ensure that the RESP scene has a SceneStop marker when no response is made, which is identified when the time limit is reached for the Stimulus Object's InputMask.

- 1) If necessary, **double click** the **Stimulus** to **open** it in the workspace.
- 2) **Click** the white **Property Pages** button.
- 3) **Select** the **Task Events** tab, and then **click** the **Add...** button.
- 4) **Double click** `Stimulus.Keyboard(1).TimeLimitReached`.

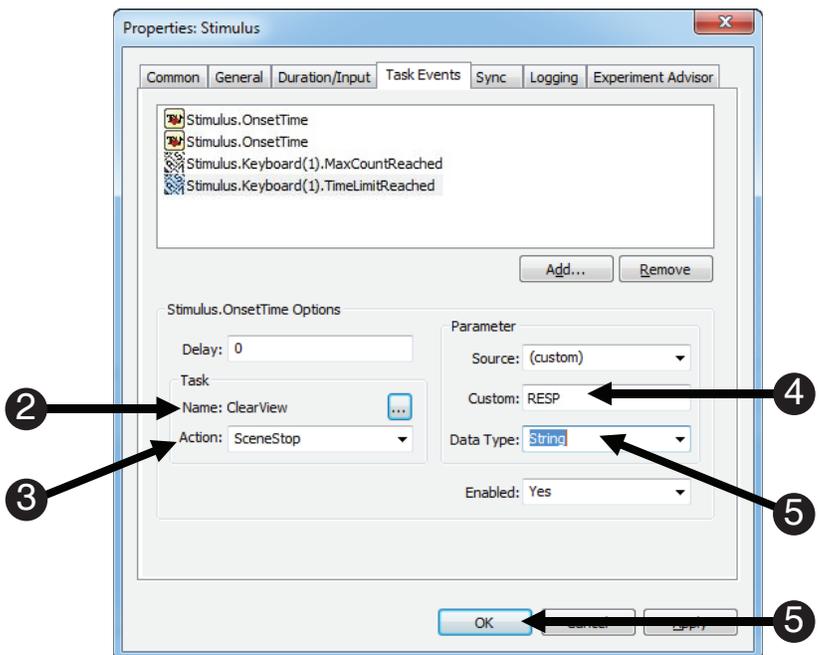
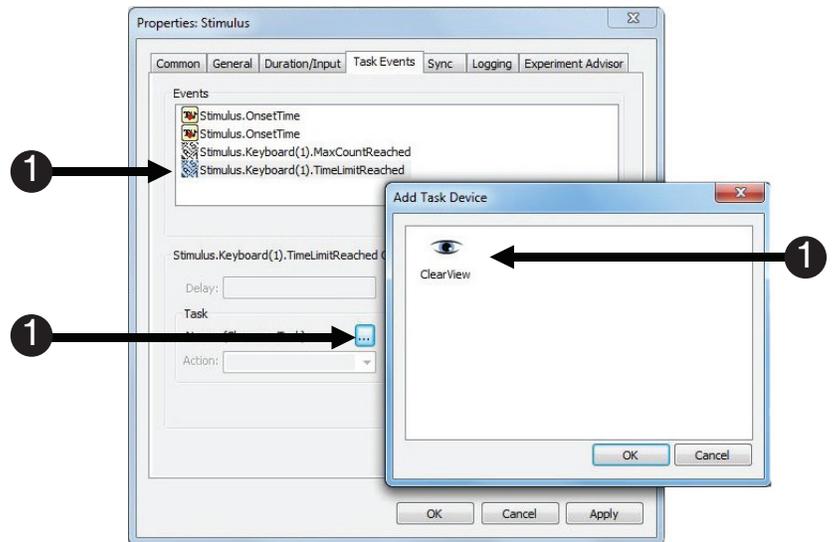


## Task 24: Edit the Task associated with the Stimulus.Keyboard(1).TimeLimitReached Event

Define the "Task" component of the Task Event as sending the RESP SceneStop marker to Tobii Studio.

This Task Event will ensure that the RESP scene has a SceneStop marker when the participant does not respond.

- 1) **Select** the **Stimulus.Keyboard(1).TimeLimitReached** task event. **Click** the ... button next to the **Name** field. **Double click** the **ClearView** icon.
- 2) **Note** the **Name** field **reads ClearView**.
- 3) **Select SceneStop** from the **Action** field.
- 4) **Edit** the **Custom** field to **read RESP**.
- 5) **Select String** as the **Data type**, and **click OK**.

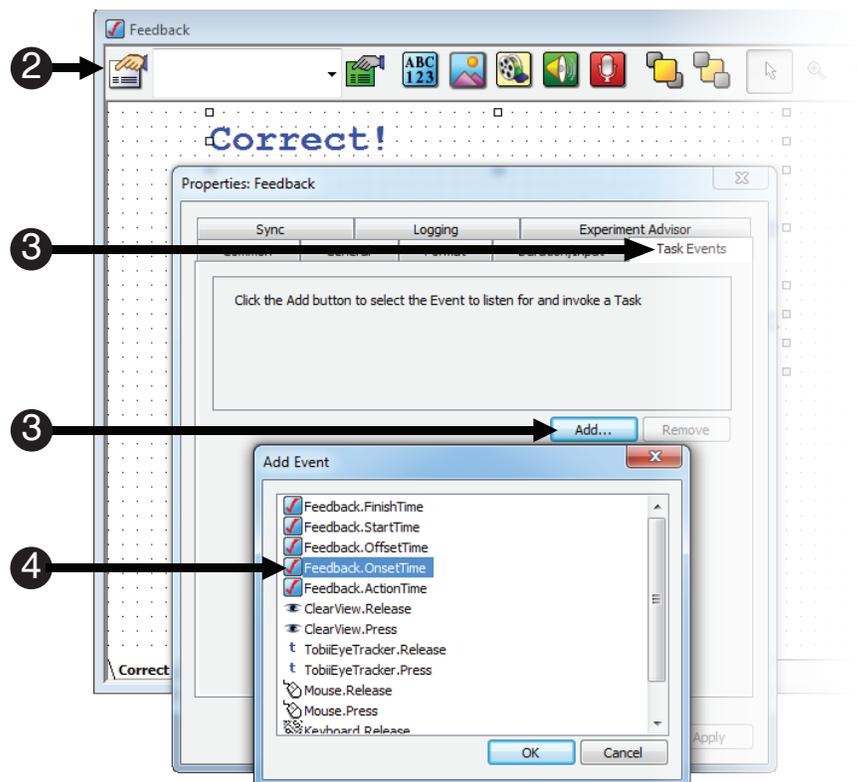
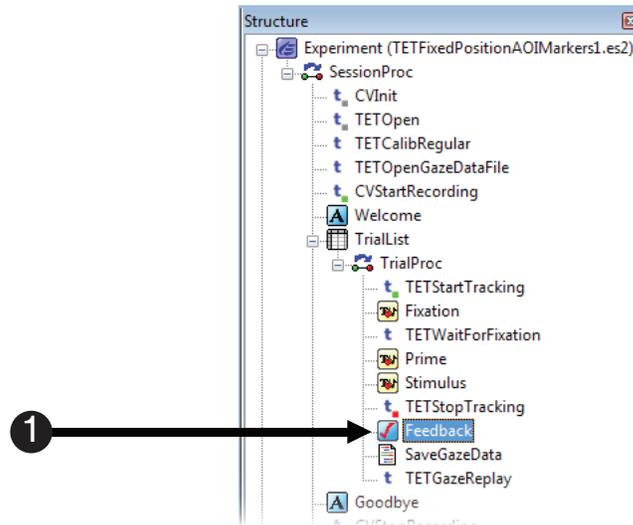


## Task 25: Add a Task Event to the Feedback Object

Add the `Feedback.OnsetTime` Task Event to the Feedback Object.

This step will add the `.OnsetTime` task event to the Feedback Object. This task event will send the end of scene marker for the Stimulus scene to Tobii Studio.

- 1) **Double click** the **Feedback** to **open** it in the workspace.
- 2) **Click** the white **Property Pages** button.
- 3) **Select** the **Task Events** tab, and then **click** the **Add...** button.
- 4) **Double click** **Feedback.OnsetTime**.

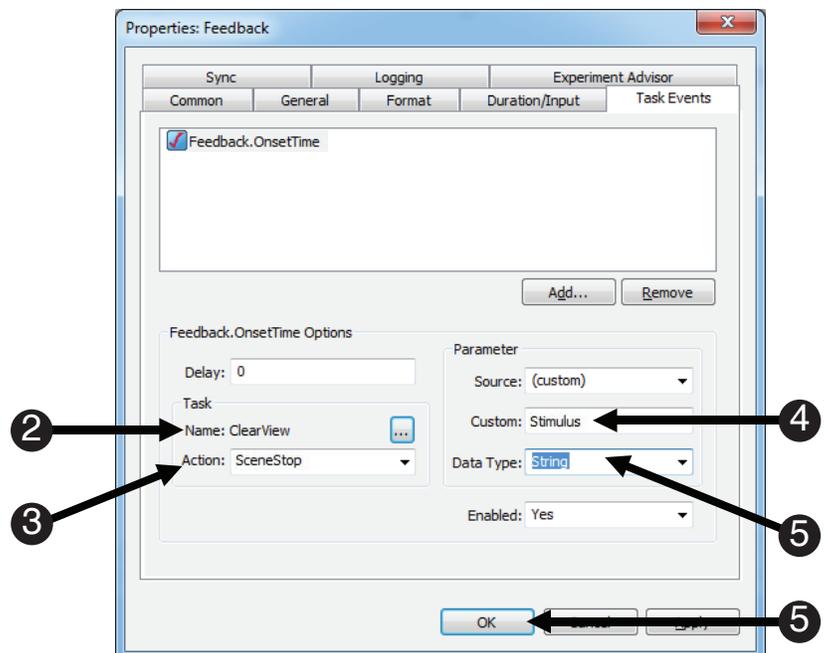
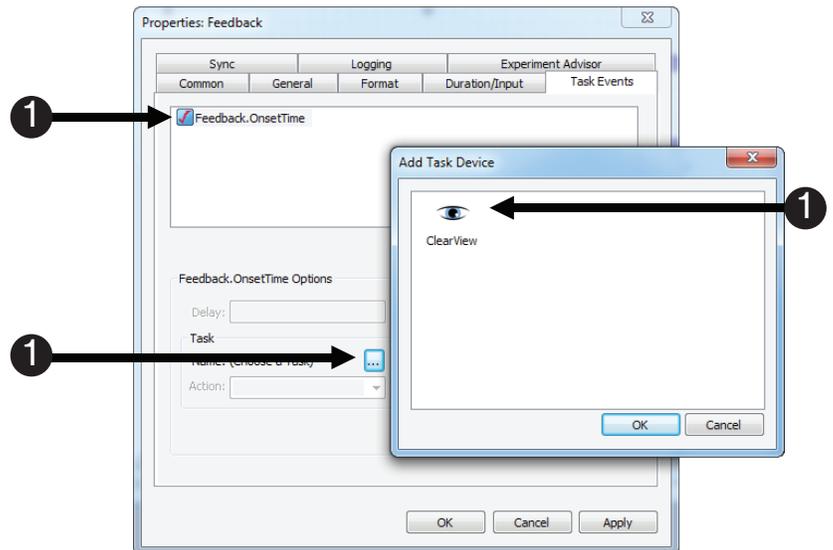


## Task 26: Edit the Feedback.OnsetTime Task Event

Define the "Event" component as Feedback.OnsetTime.

Define the "Task" component of the Task Event by sending the Stimulus SceneStop marker to Tobii Studio.

- 1) **Select** the **Feedback.OnsetTime** task event. **Click** the ... button next to the **Name** field. **Double click** the **ClearView** icon.
- 2) **Note** the **Name** field **reads** **ClearView**.
- 3) **Select** **SceneStop** from the **Action** field.
- 4) **Edit** the **Custom** field to **read** **Stimulus**.
- 5) **Select** **String** as the **Data** type, and **click** **OK**.

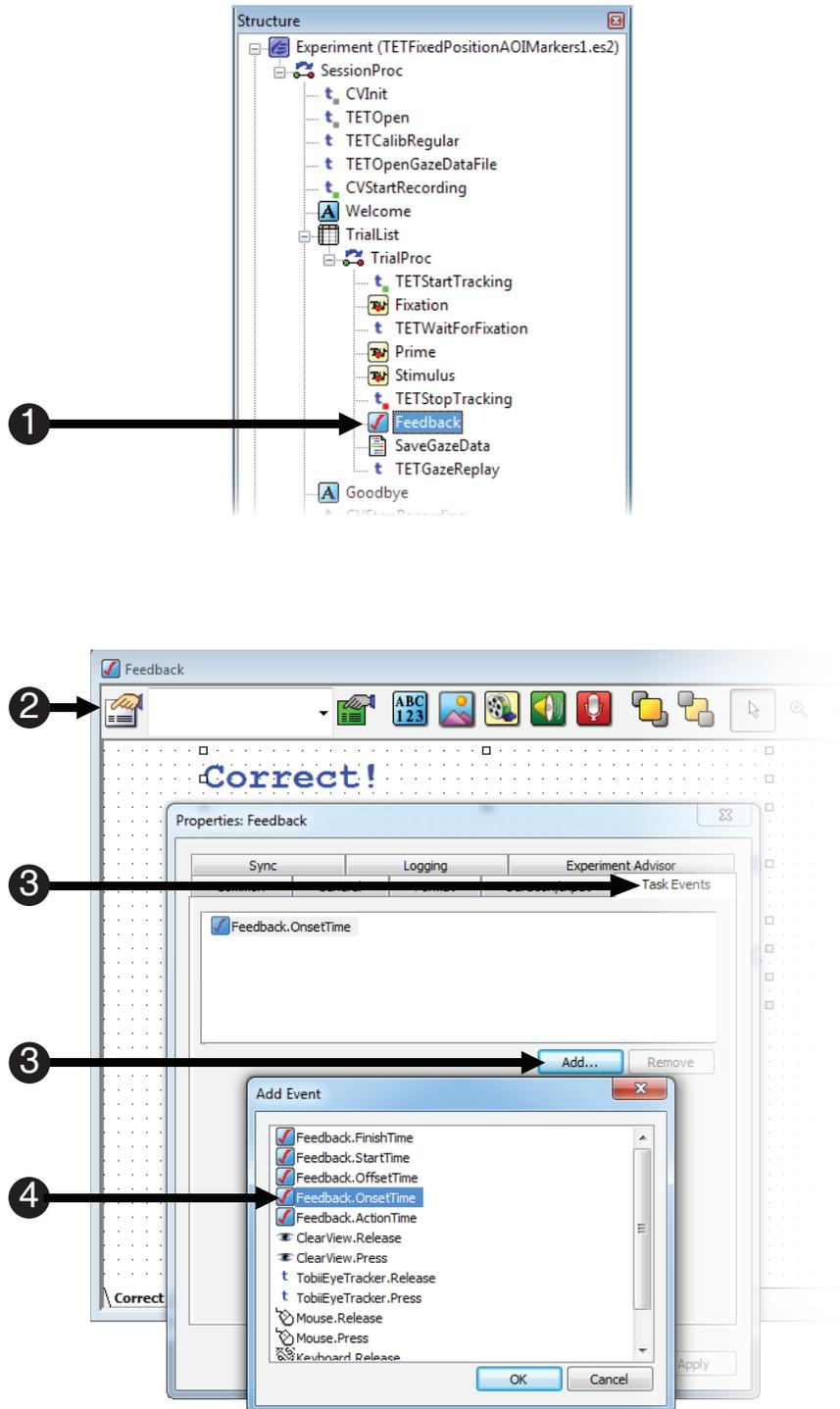


## Task 27: Add three more Task Events to the Feedback Object

Use the `Feedback.OnsetTime` event as the trigger for several scenes.

The next step is to add three more Task Events to the Feedback Object, all triggered from the `Feedback.OnsetTime` event. We use this event to mark the end of several scenes that were started earlier in this tutorial on other objects. We also create a Feedback scene. First, we end the RightImage scene.

- 1) If necessary, **double click** the **Feedback Object** to **open** it in the workspace.
- 2) **Click** the white **Property Pages** button.
- 3) **Select** the **Task Events** tab, and then **click** the **Add...** button.
- 4) **Double click** **Feedback.OnsetTime**.

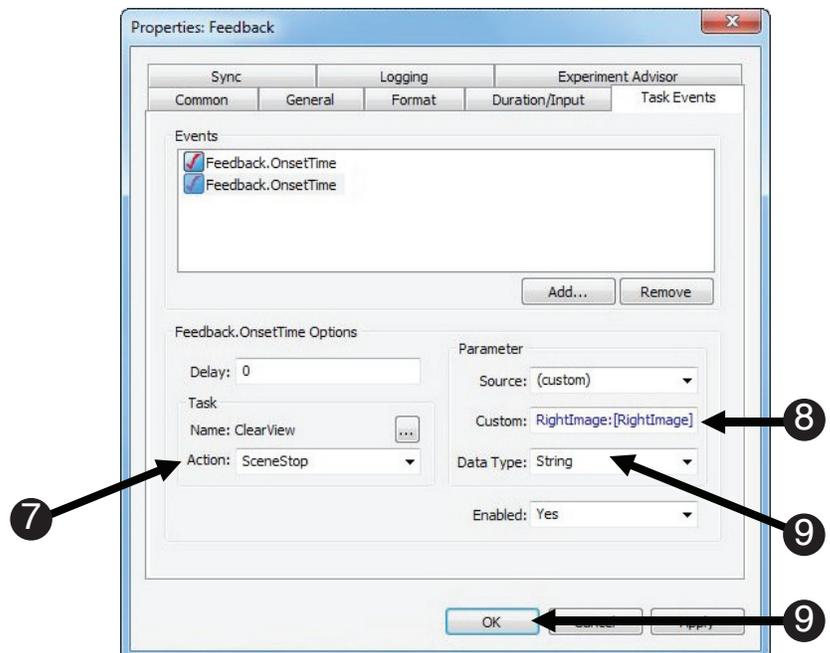
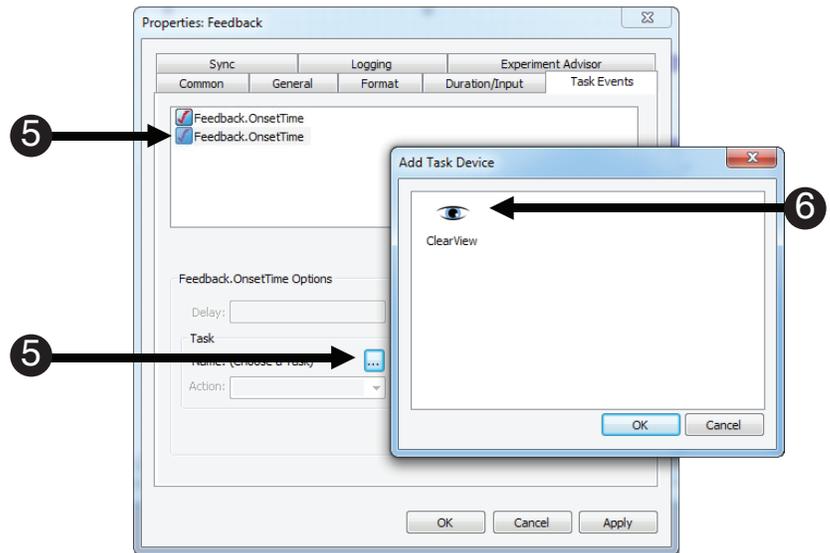


## Task 27 (continued): Add three more Task Events to the Feedback Object

Use the `Feedback.OnsetTime` event as the trigger for several scenes.

We now define the "Task" component of this second Task Event as sending the `SceneStop` marker for the `RightImage` scene to Tobii Studio.

- 5) **Select** the second **Feedback.OnsetTime** task event. **Click** the ... button next to the **Name** field. **Double click** the **ClearView** icon.
- 6) **Note** the **Name** field **reads** **ClearView**.
- 7) **Select** **SceneStop** from the **Action** field.
- 8) **Edit** the **Custom** field to **read** **RightImage:[RightImage]**.
- 9) **Select** **String** as the **Data** type, and **click** **OK**.

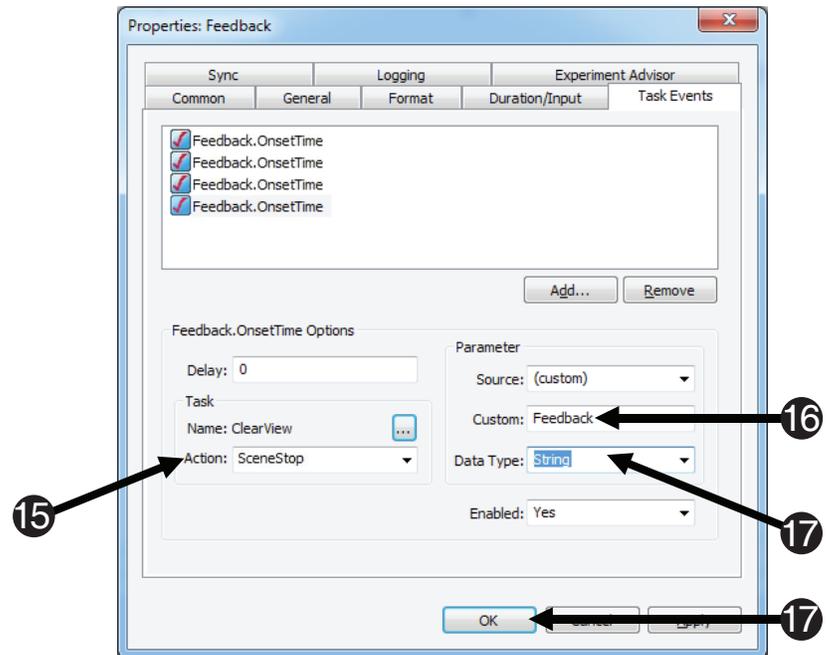


## Task 27 (continued): Add three more Task Events to the Feedback Object

Use the `Feedback.OnsetTime` event as the trigger for several scenes.

We now end the LeftImage scene. Staying with the Feedback Object and still using the `Feedback.OnsetTime` event, we also create the start of a new scene named Feedback.

- 10) **Repeat** steps 2 through 4 from earlier in this task to create a third **Feedback.OnsetTime** event.
- 11) **Select SceneStop** from the **Action** field.
- 12) **Select SceneStop** from the **Action** field.
- 13) **Select String** as the **Data** type, and **click OK**.
- 14) **Repeat** steps 2 through 4 from earlier in this task to create a fourth **Feedback.OnsetTime** event.
- 15) **Select SceneStop** from the **Action** field.
- 16) **Edit** the **Custom** field to **read Feedback**.
- 17) **Select String** as the **Data** type, and **click OK**.

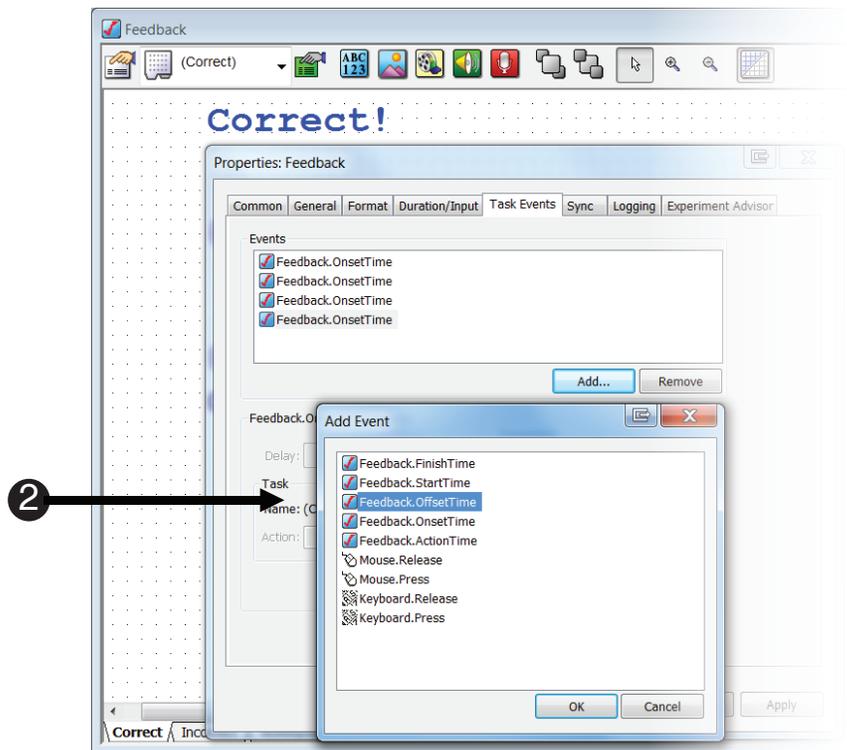


## Task 28: Add another Task Event to the Feedback Object

Define the "Event" component of the Task Event with the `Feedback.Offset` as the trigger.

This step adds the `Feedback.OffsetTime` event as the triggering event. This event serves as the end of the Feedback scene.

- 1) **Repeat** steps 2 through 4 from **Task 27** to add another Task Event to **Feedback Object**.
- 2) **Double click** `Feedback.OffsetTime`.

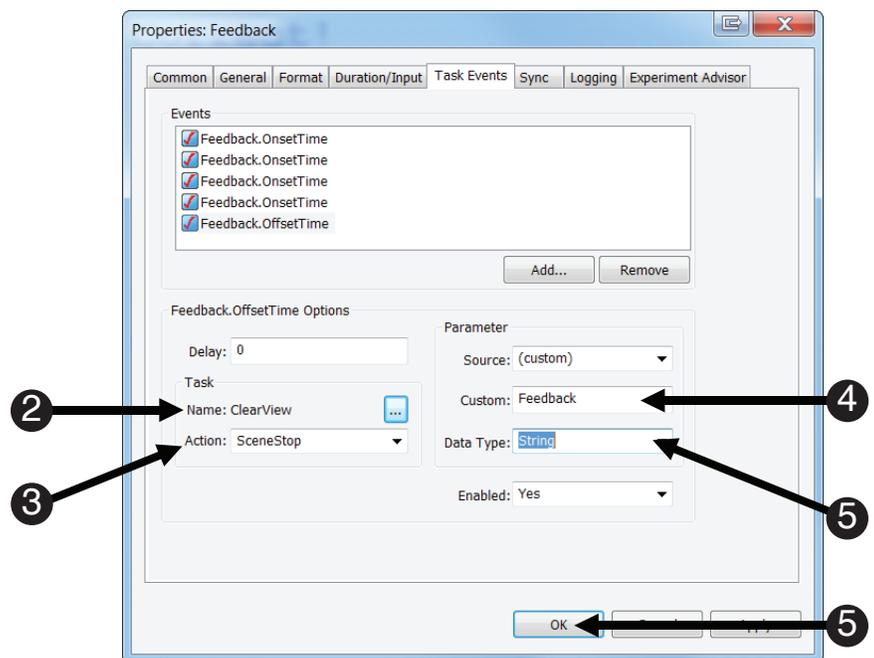
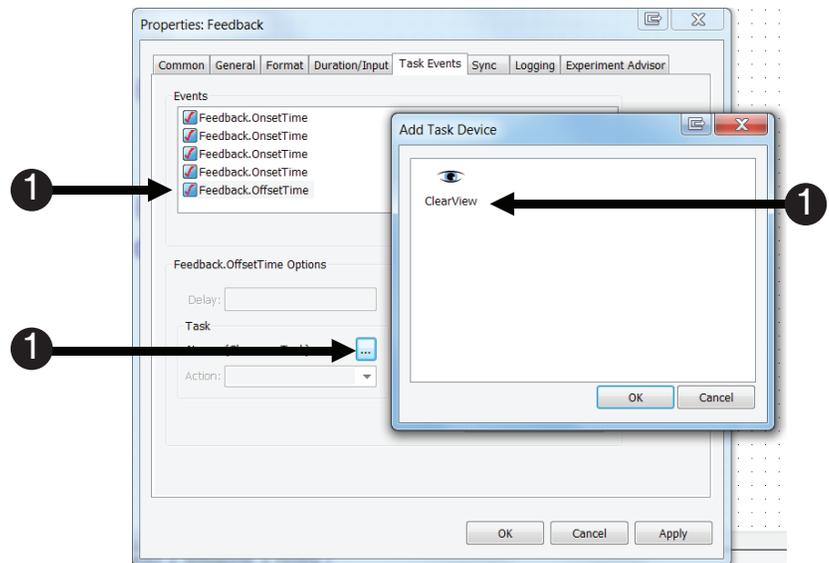


## Task 29: Complete the Task Event for the Feedback Object

Define the "Task" component of the Task Event as sending the Feedback SceneStop marker to Tobii Studio.

This step sends the SceneStop marker for the Feedback Object.

- 1) **Select** the **Feedback.OffsetTime** task event that you created in the previous task.  
**Click** the ... button next to the **Name** field. **Double click** the **ClearView** icon.
- 2) **Note** the **Name** field **reads** **ClearView**.
- 3) **Select** **SceneStop** from the **Action** field.
- 4) **Edit** the **Custom** field to **read** **Feedback**.
- 5) **Select** **String** as the **Data** type, and **click** **OK**.

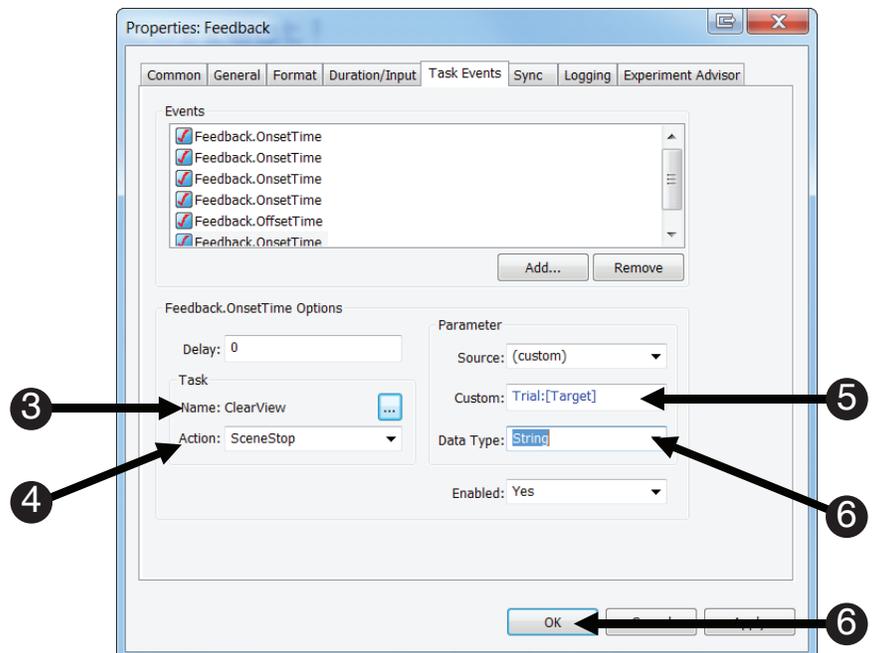
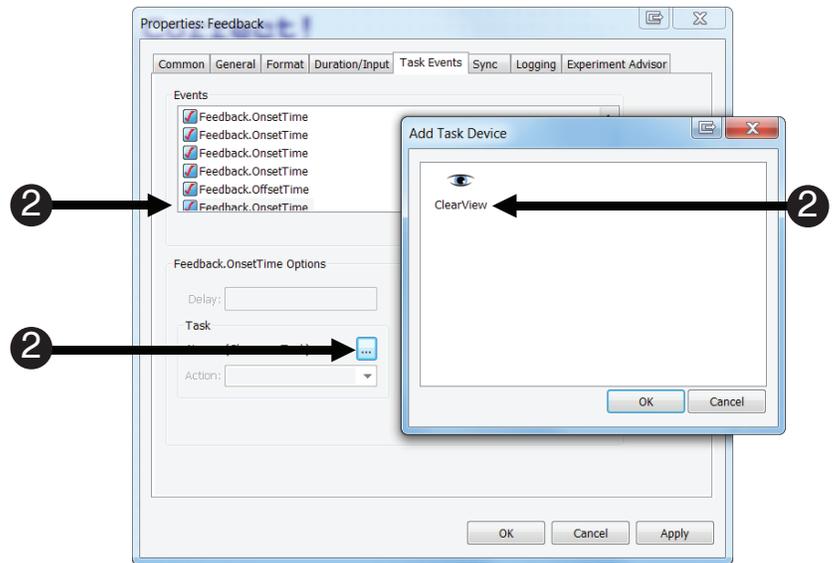


## Task 30: Create the final Task Event to end the Trial:[Target] scene

Configure one more Task Event on the Feedback Object to end the Trial scene.

This step sends the SceneStop marker for the Trial:[Target] scene. This is the last SceneStop marker that needs to be defined.

- 1) **Repeat** steps 2 through 4 from **Task 27** to add another Task Event to **Feedback Object**.
- 2) **Select** the **Feedback.OnsetTime** task event. **Click** the ... button next to the **Name** field. **Double click** the **ClearView** icon.
- 3) **Note** the **Name** field **reads** **ClearView**.
- 4) **Select** **SceneStop** from the **Action** field.
- 5) **Edit** the **Custom** field to **read** **Trial:[Target]**.
- 6) **Select** **String** as the **Data type**, and **click** **OK**.

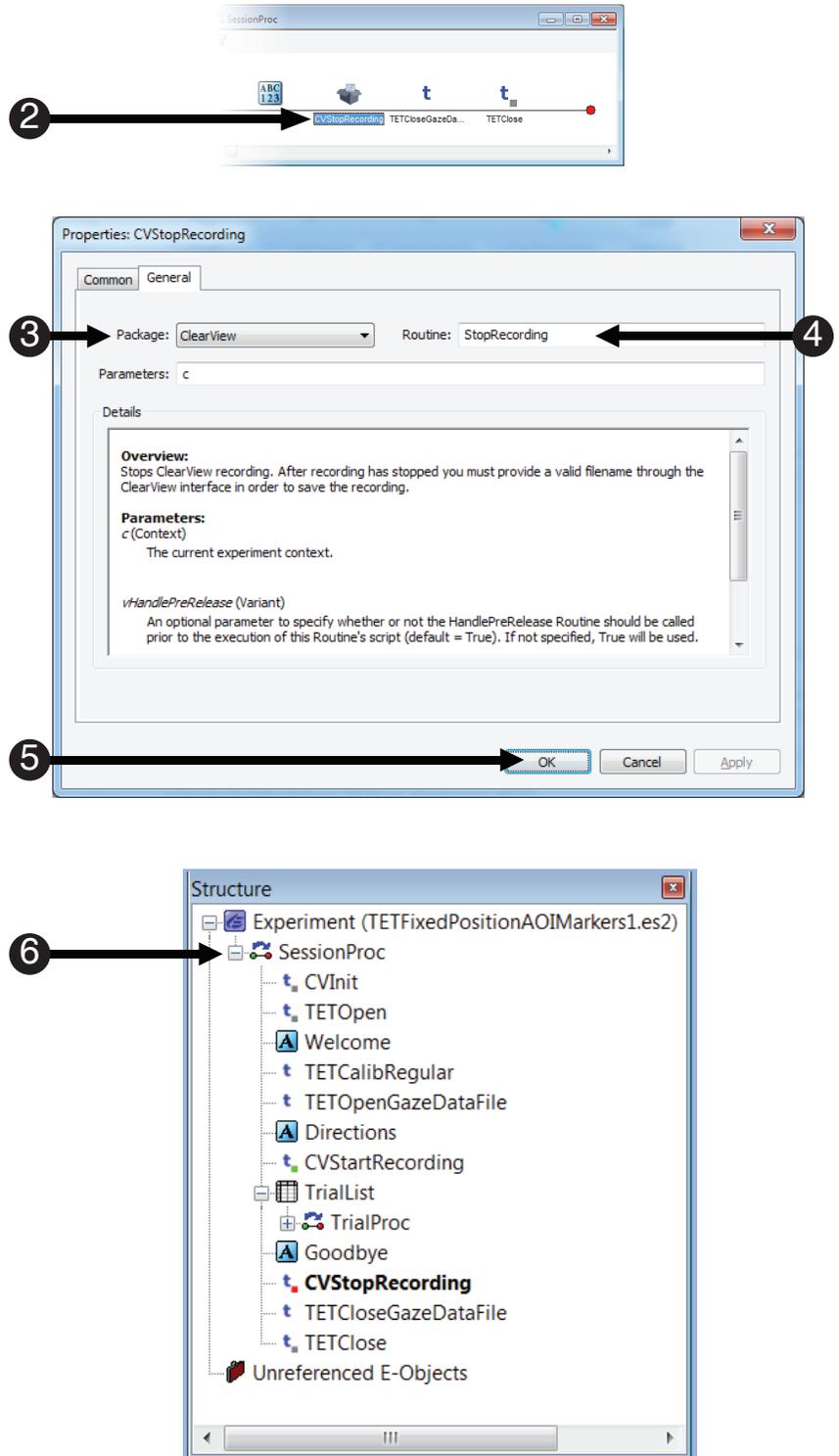


## Task 31: Add the CVStopRecording PackageCall

Add a PackageCall to the SessionProc and name the PackageCall “CVStopRecording,” and accept the default parameters.

The CVStopRecording PackageCall stops ClearView from recording. This PackageCall should be placed at the location in the procedure where you want to stop communication with Tobii Studio. Accept the default parameters.

- 1) **Double click** the **SessionProc** to open it. **Drag** and **drop** a new **PackageCall** from the **Toolbox** **before** the **TETCloseDataFile PackageCall** and **after** the **Goodbye Object**. **Rename** the **PackageCall** to “**CVStopRecording**” and **press Enter** to accept the change.
- 2) **Double click** the **CVStopRecording Object** on the **SessionProc** to display its **Property Pages**.
- 3) **Select ClearView** from the **Package** dropdown list.
- 4) **Select StopRecording** from the **Routine** dropdown list.
- 5) **Click** the **OK** button to accept the default parameters.
- 6) **Confirm** your **SessionProc** is identical to the example shown.

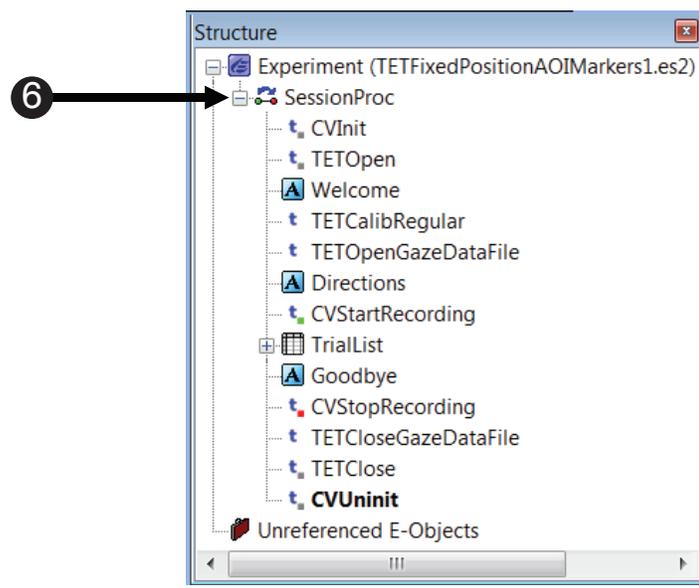
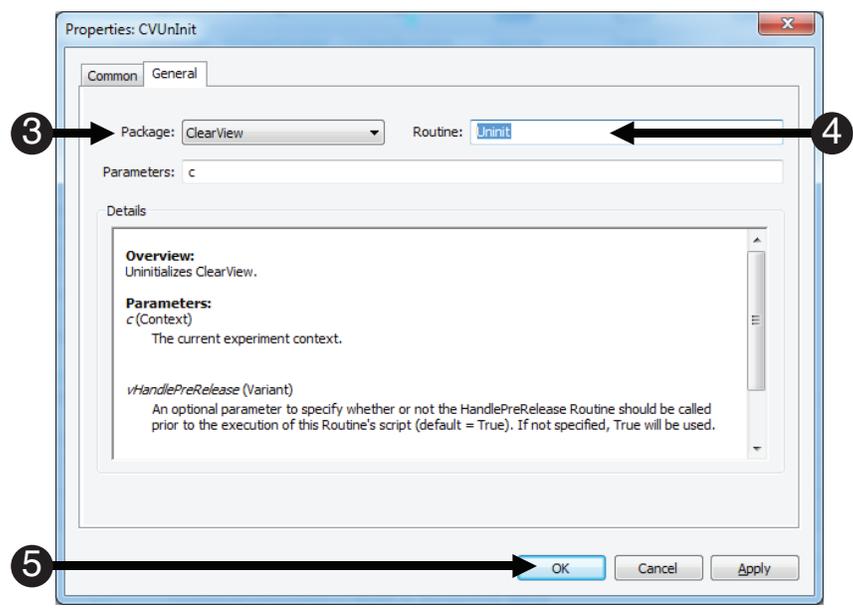


## Task 32: Add the CVUnInit PackageCall to uninitialized the ClearView Packages and accept the default parameters

Add a PackageCall to the SessionProc and name the PackageCall "CVUnInit."

The CVUnInit PackageCall uninitialized the ClearView package file. Add the PackageCall to the SessionProc and Accept the default parameters.

- 1) **Drag and drop** a new **PackageCall** from the **Toolbox** after the **TETClose PackageCall**. **Rename** the **PackageCall** to "**CVUnInit**" and **press Enter** to accept the change.
  - 2) **Double click** the **CVUnInit Object** on the **SessionProc** to display its **Property Pages**.
  - 3) **Select ClearView** from the **Package** dropdown list.
  - 4) **Select UnInit** from the **Routine** dropdown list.
  - 5) **Click** the **OK** button to accept the default parameters.
  - 6) **Confirm** your **SessionProc** is identical to the example shown.
- NOTE:** You will need to set up an external video object in Tobii Studio to collect data from this experiment. Please refer to Tobii Studio documentation on how to do this.



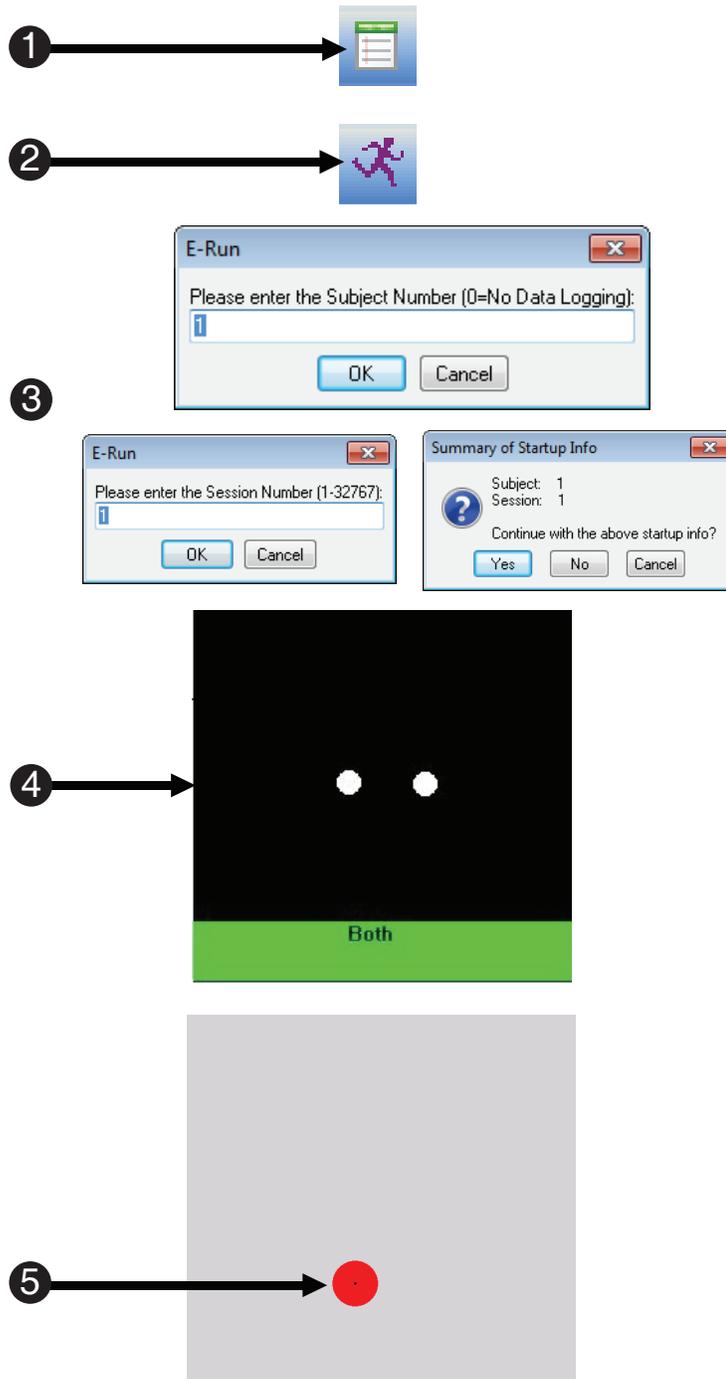
## Task 33: Run the Experiment

Run the experiment to verify that the SceneStart and SceneStop markers are being sent by E-Prime and received by Tobii Studio as expected.

**⚠ NOTE:** As was noted in the introduction to this tutorial, Tobii Studio requires that calibration be performed to start the external video object. Since this experiment already includes calls to the TET PackageFile Routine that performs calibration, you will end up performing calibration twice. If you wish to avoid the calibration step in E-Prime, highlight the TETCalibRegular PackageCall and press Delete. This will move the PackageCall to Unreferenced E-Objects, and allow the experiment to be run without first performing the extra calibration.

The next steps will walk you through generating the experimental script, starting the experimental paradigm, calibration, and running the experiment.

- 1) **Press Ctrl+S** to save your work before continuing. **Click** the **generate icon** or **press Ctrl+F7** to generate the script and **check** it for **errors**.
- 2) **Click** the **run icon** or **press F7** to **run** the paradigm.
- 3) **Click OK** to accept the **default values** for **Subject Number**, **Session Number** and **Yes** for **Summary of Startup Info**.
- 4) **Look** at the **screen** to **verify** the **eyes** are **stable** in the track status window. **Ensure** that there are **two white dots** that appear in the box and that the **bottom bar** is **green**. **Press Space** once both eyes are stable.
- 5) **Run** through the **calibration process** by **following** the **dots** on the **screen** with your **eyes**.



## Task 33 (continued): Run the Experiment

Run the experiment to verify that the eye tracker is working, and to ensure the .gazedata file is written.

Once you have completed calibration, you will be prompted to verify that the data is acceptable.

6) **Accept the calibration.**

**⚠ NOTE:** Please refer to Tutorial 1, Task 17 (continued): Run the Experiment, (Page 44) for additional information about calibration.



# Scene-based Analysis of E-Prime Experiments in Tobii Studio

---

Tobii Studio supports segment-based and scene-based analysis of data sent from E-Prime. For more information on segment-based analysis, see the [KB 5624: INFO: Supporting Segment-based Analyses in E-Prime Extensions for Tobii](#). This section focuses on Scene-based analysis. The advantage of a scene-based analysis in Tobii Studio is that it provides more robust data analysis based on E-Prime scenes. Scene-based analysis affords the opportunity to receive more descriptive scene data analyses sent over by E-Prime during a trial run using the Task Events feature.

During this tutorial, you created an E-Prime experiment that defined multiple scenes. The purpose of these scenes is to send information about the experiment trial sequence to Tobii Studio. Tobii Studio, in turn, can utilize the scene information to perform complex analysis, including the creation and display of gaze visualization and heat maps.

Once you have successfully run your experiment in E-Prime, please switch to the Tobii computer. You should no longer see the same “record” screen that you did before the run of the experiment. This means that the run has been completed and that there should be data.

This section describes the steps that we took to perform a meaningful analysis on the data that is created from the completed Tobii Studio experiment in the previous tutorial. With this data, we are interested how long participants looked at both the Distractor and Stimulus images during the entire experiment. This will allow us to determine if the participant was spending more time fixating on the distractor image in each task or the target stimulus for each task. To get this information, we will look at two scenes that we created in the previous tutorial. While the other scenes are useful, these are the only two that we will focus on in this chapter. Unlike the previous sections of this tutorial, we do not present a step-by-step guide to using Tobii Studio here. Instead, this section is intended to provide a very high-level overview of analyzing gaze data in Tobii Studio. The steps listed below are meant only as a guide and in some instances do not reflect the order in which events need to be carried out. This section demonstrates how to pull meaningful data from the scenes that were created in E-Prime as well as give a brief overview of the many data analysis options available for the gaze data. For specific information on data analysis in Tobii Studio, please refer to the [Tobii Studio User Manual](#).

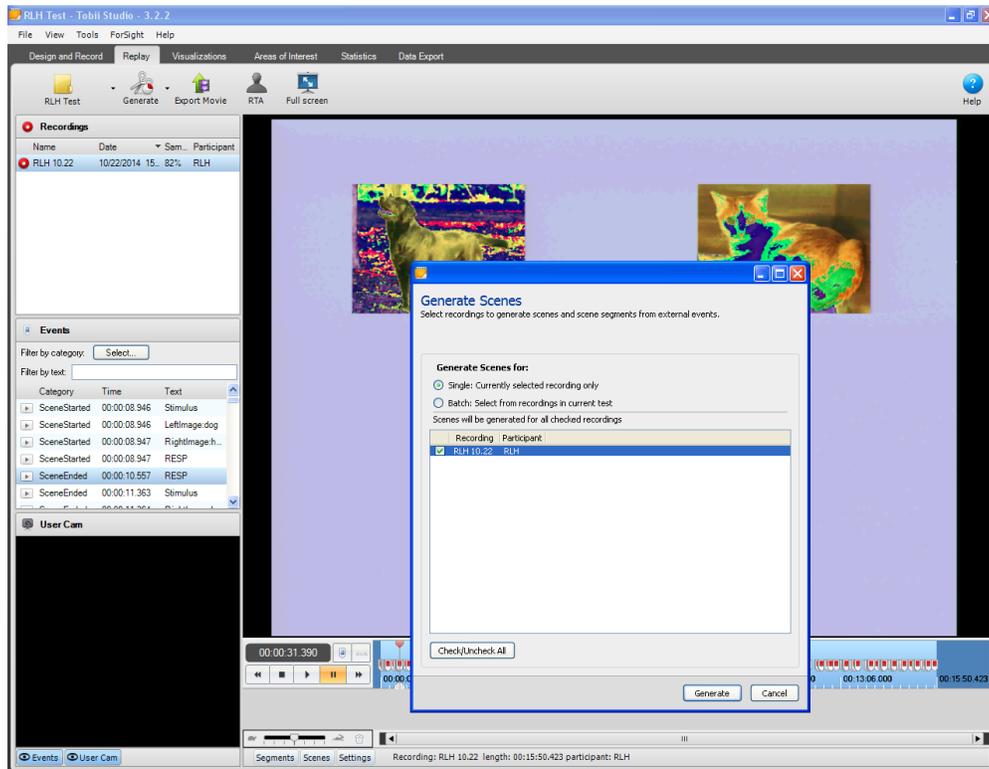
## 1) Compile and Generate Scene Data

Selecting the Replay Tab allows the user to access basic statistics about the participant’s data file. Minimally, there will be a full listing of all of the pre-defined scenes in the Events window on the left-hand side of the window. At the bottom of the screen, you will see an Event Viewer. This contains a timeline of the entire experiment with a red flag in the timeline that represents the SceneStart and SceneStop markers that were sent to Tobii during the recording.

In order to perform the aforementioned analysis on our scene data, we have to generate our scenes in Tobii Studio. The purpose of this is to pair each SceneStart marker with every SceneStop marker of the same name to create a cohesive scene.

Please note that every SceneStart marker needs a SceneStop marker in order to create a Scene of any particular event. **⚠️ NOTE:** *If a scene is missing either a SceneStart or SceneStop marker, Tobii Studio will not create a scene of that event.*

To generate the scenes in Tobii Studio, select Generate under the Replay tab, select Scenes, and then select the recordings for which you would like to analyze scenes:



## 2) Review Visualizations (Optional)

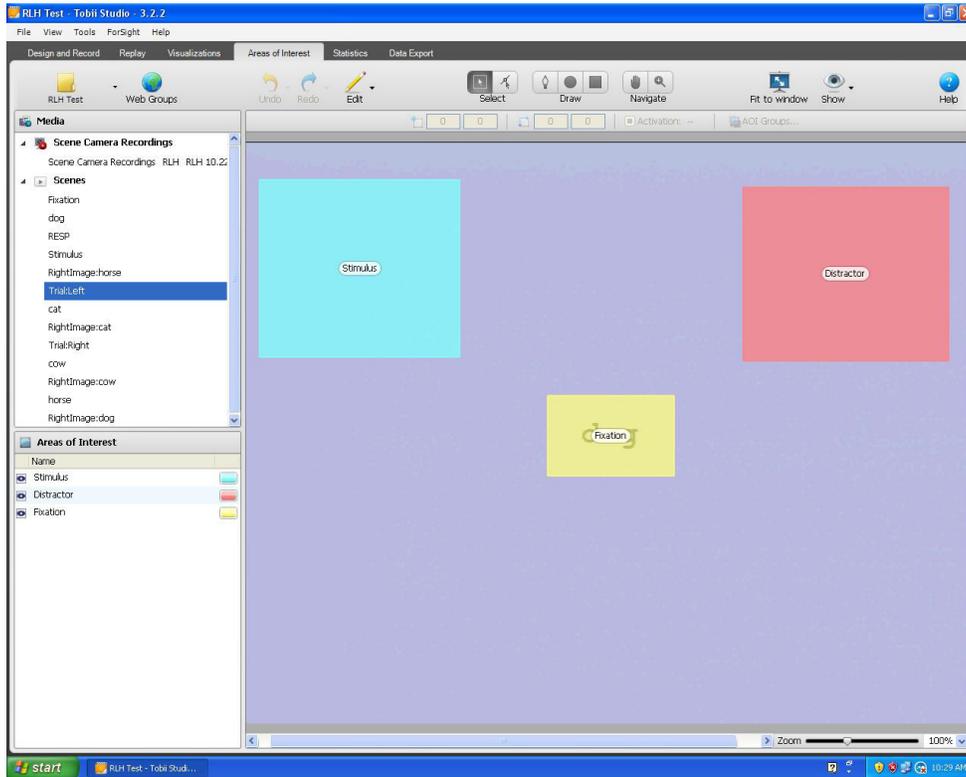
The Visualization Tab provides a variety of visual representations of the gaze data for each scene. Gaze Plot, Heat Map and Gaze Cluster analyses are supported. While this data is of interest to many researcher studies, it is not relevant to the analysis that is described here and therefore we will not be working within this tab.

## 3) Create Areas of Interest (AOIs)

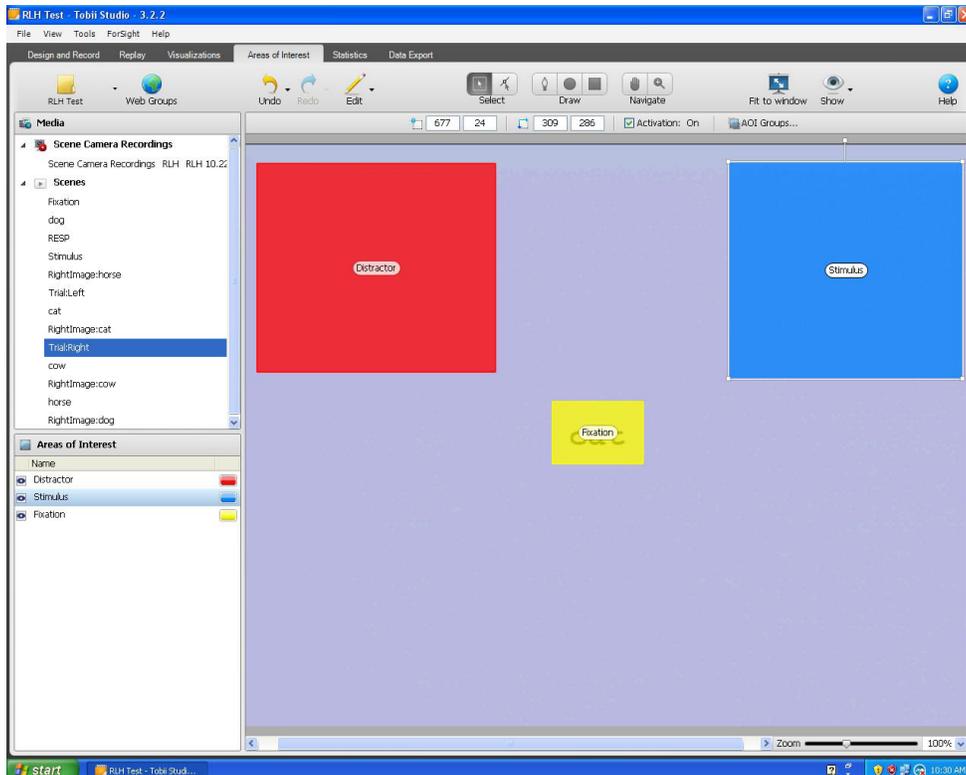
One of the most important steps of analyzing scenes in Tobii Studio is to create Areas of Interest (AOIs) for each scene. AOIs allow Tobii Studio to produce more robust data by defining areas of the screen considered to be the important or interesting parts of each scene.

To do this, first click on the Areas of Interest Tab. Next, select the scene from which you would like to analyze data. Click on the “Draw” button at the top of the window then drag and drop the areas of the screen of which you would like to make an AOI. Once the AOI is created, name the AOI something descriptive.

For this analysis, first click on the Trial:Left scene. We will need to create three AOIs here. The first AOI will cover the location of the distractor image as it appears on the screen. This AOI will be called “Distractor”. Since this is the Trial:Left condition, the distractor image will always be on the right. The next AOI will cover the target stimulus object as it appears on the screen. Since this is the Trial:Left condition, the target Stimulus will always appear on the left side of the screen. Name this AOI “Stimulus”. Finally, place a small AOI in the very center of the screen; this AOI will be the fixation area of interest, called “Fixation”. This Fixation AOI will not be directly used in our analysis as we are only interested in the distractor and target stimulus. However, adding the Fixation as an AOI is still very helpful, especially if further analysis of this data is planned.



The AOI scheme that is now placed on the Trial:Left scene needs to be repeated on the “Trial:Right” scene. This can be done by simply highlighting the AOIs on the Trial:Left scene object in Tobii Studio, pressing Ctrl+C to copy, opening up the Trial:Right scene and pressing Ctrl+V to paste these objects. This will allow you to automatically create the necessary AOIs. Keep in mind that the location of the Distractor and Stimulus AOIs will have to change for this scene. This is just a simple matter of renaming and re-coloring the AOIs.



## 4) Display Statistics

To have Tobii Studio generate the statistics from the AOIs that were created in Step 4 above, click on the Statistics tab at the top of the window. In this tab, select the Trial:Left and Trial:Right scenes. They should be the only scenes that you are able to add because they are the only scenes that contain AOIs.

After making this selection, the screen on the right-hand side of the window will require you to update it. Click on the Update button to display the newly created statistics. Before clicking on this button, locate and select the Metrics button at the top of the screen. In the Metrics dropdown you will see that Time to First Fixation is already checked. Click on this once to remove its checkmark. Once you have done this, scroll down until you see the Visit Duration option and select this. If you would like to add additional statistics to your analysis, click on the Metrics button at the top of the window and select the analysis that best suits your paradigm. Now that the Visit Duration metric is selected, click the update button.

The data can be displayed in either a chart format or in a table format. Both formats are demonstrated below. This data will now tell us exactly how long participants looked at the distractor and the target stimuli for the experiment.

