E-Prime
Extensions for
tobii pro

User Manual

For Research and Education Only

Psychology Software Tools, Inc.
311 23rd Street Extension, Suite 200
Sharpsburg, PA 15215-2821
Phone 412.449.0078
Fax: 412-449-0079
Email: info@pstnet.com

E-Prime Extension for Tobii Pro 3.2 User Manual
PST-101744
Rev 6

Copyright

Psychology Software Tools, Inc.
311 23rd Street Extension, Suite 200
Sharpsburg, PA 15215-2821
Phone: 412-449-0078
Fax: 412-449-0079
E-mail: info@pstnet.com
Web: www.pstnet.com

For questions or comments regarding this manual or installation assistance:
Please e-mail us at support@pstnet.com or visit us at https://support.pstnet.com

Trademark

This manual describes the installation procedure for the E-Prime Extensions for Tobii Pro 3.2. Please review the manual completely and thoroughly before beginning the system installation.

E-Prime Extensions for Tobii Pro 3.2 Software is for research and educational purposes only.

Table of Contents

1. Getting Started 6

1.1. Product Overview 6

1.2. System Requirements 6

1.3. Product Service and Support 6

1.4. Resources 7

2. Installing E-Prime Extensions for Tobii Pro 8

2.1. Before Installation 8

2.2. During Installation 9

2.3. After Installation 12

3. How to Create Your Own Tobii Pro Experiment 14

3.1. Overview 14

3.2. Experiment Design 14

3.3. Tutorial 1: Adding Tobii Pro Support to an E-Prime Experiment 18

3.4. Tutorial 2: Logging Gaze Data in an Experiment 44

3.5. Tutorial 3: Multiple Monitors: Creating a Participant Station 60

3.6. Tutorial 4: Working with Eye Gaze Data Interactively 70

3.7. Tutorial 5: Adding Tobii Pro Lab Support to an Experiment 78

4. Troubleshooting 107

4.1. Cannot locate E-Prime Samples and Tutorials 107

4.2. Running without an eye tracker 108

4.3. Missing and Incomplete Parameter Errors 109

4.4. Unable to Open/Connect to the Eye Tracker 111

4.5. Upgrading Experiments to E-Prime Extensions for Tobii 112

4.6. Locate Your IP Address and Ping Another Computer 113

4.7. Issues with Improperly Configuring Tobii Pro Lab 115

Appendix A: Samples and Tutorials 116

Appendix B: Hardware Configurations 117

B.1 Hardware Configuration with One Machine 118

B.2 Hardware Configuration with Two Machines (Calibration in E-Prime – Preferred) 118

B.3 Hardware Configuration with Two Machines (Calibration in Tobii Pro Lab) 119

Appendix C: Timing and Synchronization: Writing to the Buffer and Buffer Size 120

5. Contact Information 121

1. Getting Started

1.1. Product Overview

E-Prime Extensions for Tobii Pro 3.2 (EET) is a set of software routines that allow communication between the TET Server (Tobii Pro Eye Tracker Server) and E-Prime during experiments. It allows you to create E-Prime experiments or update existing E-Prime experiments to function with Tobii Pro Eye Trackers. These experiments are referred to throughout the remainder of this manual as TET-enabled experiments. EET 3.2 also allows communication between E-Prime and the Tobi Pro Lab software package for visualization and analysis of eye tracking data. These experiments are referred to throughout the remainder of this manual as TPL-enabled experiments. The extensions also include a set of sample experiments that can be run directly or used as a basis from which to create new experiments.

1.2. System Requirements

E-Prime Extensions for Tobii Pro 3.2 is compatible with ***E-Prime 3.0 Update 2***. E-Prime 2.0 is not supported with the current software. The E-Prime 3.0 file extension is .es3. This manual uses the .es3 extension. Prior to the E-Prime Extensions for Tobii Pro 3.2 installation, you need to determine which version of E-Prime you currently have on your machine.

NOTE: *If permitted by your research institution, we recommend updating to the latest version of E-Prime for new features and bug fixes.*

Software Compatibility Requirements

E-Prime 3.0 and E-Prime Extensions for Tobii Pro 3.2 are compatible with machines running Microsoft® Windows® 10.

Computer Requirements for the machine running E-Prime

Additionally, the computer must comply with the minimum E-Prime 3.0 Update 2 requirements:

E-Prime 3.0 Update 2 Requirements

NOTE: *For further computer requirements for the Tobii Pro software and Tobii Pro Eye Tracker, please refer to the Tobii Pro documentation.*

1.3. Product Service and Support

Technical Support

Psychology Software Tools, Inc. provides technical support for E-Prime via the PST Product Service and Support website. To receive technical support, must register online at:

https://support.pstnet.com/

Registration requires a valid E-Prime serial number, an E-Prime Extensions for Tobii Pro 3.2 serial number, and an e-mail address. A Knowledge Base is available on our Product Service and Support website with release notes and frequently asked questions. The site includes E-Prime sample paradigms available for download, How-To videos, and the E-Prime 3.0 Experiment Library.

Our Product Service and Support website offers an additional support forum used to post general questions about E-Prime and E-Prime Extensions for Tobii Pro 3.2:

Psychology Software Tools Community Forum

Additionally, Psychology Software Tools has a YouTube channel with various Live Stream Webinars and How-To videos:

PSTNET on YouTube

NOTE: *This document contains links to our online Knowledge Base which has articles for E-Prime Extensions for Tobii Pro 3.2. If there is trouble opening the link, please search the article number on our Product Service and Support website.*

1.4. Resources

Tobii Pro Resources

- Tobii provides technical support for Tobii Pro related hardware and software:

  Tobii Pro Technical Support

- Tobii Pro product pages:

  Tobii Pro Products

Psychology Software Tools Resources

- E-Prime Extensions for Tobii Pro product page:

  EET 3.2 Product Page

- E-Prime Extensions for Tobii Pro download link:

  EET 3.2 Download Link

NOTE: *To access downloads, users must register E-Prime and E-Prime Extensions for Tobii Pro 3.2.*

2. Installing E-Prime Extensions for Tobii Pro

2.1. Before Installation

E-Prime Extensions for Tobii Pro 3.2 is compatible ***with E-Prime 3.0 only***. Prior to installing, determine which version of E-Prime exists on your machine.

For installation, you need either the EET 3.2 USB Flash Drive or the installation download that we host on our website. If using the USB Flash Drive, navigate in Windows Explorer and run the E-Prime Extensions for Tobii Pro 3.2 Setup.exe file. If using an installation download provided by our website, navigate to the Psychology Software Tools Support page and click on the download link (See Chapter 1: Section 1.4: *Resources*).

After you have determined if the correct version of E-Prime is installed on your machine, you can install E-Prime Extensions for Tobii Pro 3.2. Before continuing, be sure that you have administrative rights to install the software on your machine. The software cannot be installed without administrative rights. If you are unaware of your rights, please contact your system administrator.

If during the installation process you encounter issues that prohibit installation, contact your system administrator or contact PST's Product Service and Support at:

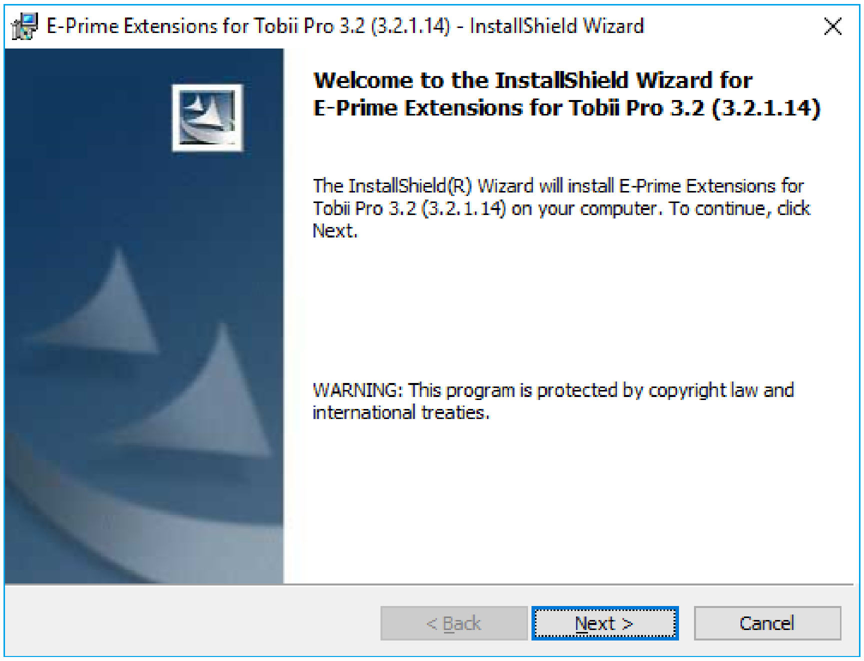
https://support.pstnet.com/

NOTE: *Please retain the packaging for your EET 3.2 Installation; it contains a label with your serial number which is required for installation. Additionally, register your serial number on our Product Service and Support website. Once your serial number is registered you can access our Downloads section of our website.*

2.2. During Installation

NOTE: *The version number on the following images may not correspond to the version you are installing.*

1) ***Insert*** the **E-Prime Extensions for Tobii Pro 3.2 USB Flash Drive** into your USB port. If the Setup program does not automatically start, ***navigate*** to your **USB drive** in Windows Explorer and ***double click*** **Setup.exe** to launch the installer. ***Click*** **Next** to continue after the InstallShield Wizard begins.



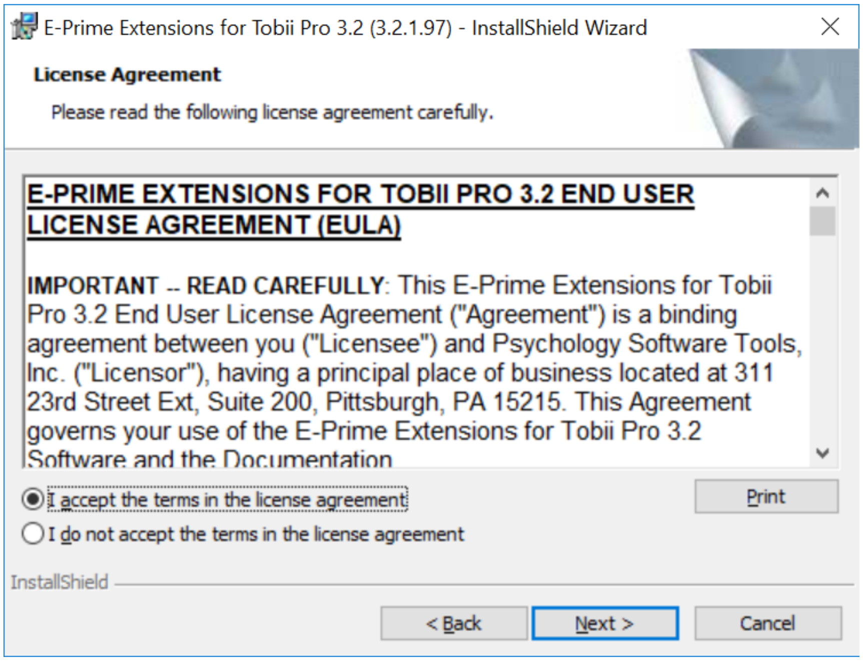
E-Prime Extensions for Tobii Pro 3.2 (3.2.1.14) - InstallShield Wizard

Welcome to the InstallShield Wizard for E-Prime Extensions for Tobii Pro 3.2 (3.2.1.14)

The InstallShield(R) Wizard will install E-Prime Extensions for Tobii Pro 3.2 (3.2.1.14) on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

2) ***Read***, and *if agreed upon*, ***Accept*** the **Terms and Conditions**. ***Click*** **Next** to continue.



E-Prime Extensions for Tobii Pro 3.2 (3.2.1.97) - InstallShield Wizard

License Agreement

Please read the following license agreement carefully.

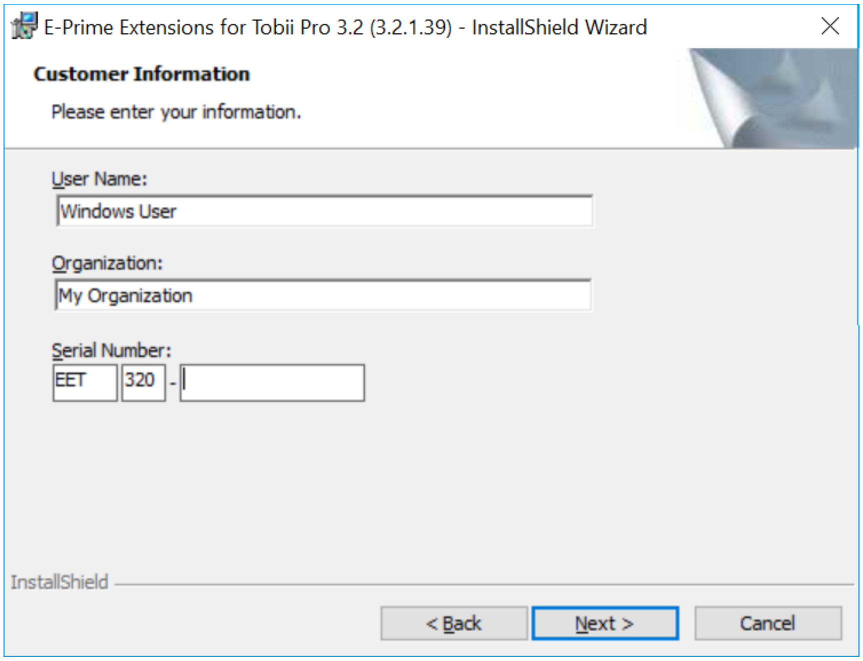
E-PRIME EXTENSIONS FOR TOBII PRO 3.2 END USER LICENSE AGREEMENT (EULA)

IMPORTANT -- READ CAREFULLY: This E-Prime Extensions for Tobii Pro 3.2 End User License Agreement ("Agreement") is a binding agreement between you ("Licensee") and Psychology Software Tools, Inc. ("Licensor"), having a principal place of business located at 311 23rd Street Ext, Suite 200, Pittsburgh, PA 15215. This Agreement governs your use of the E-Prime Extensions for Tobii Pro 3.2 Software and the Documentation

● I accept the terms in the license agreement

○ I do not accept the terms in the license agreement

3) ***Specify*** a **User Name, Institution,** and **Serial Number**. ***Click*** **Next** to continue.



E-Prime Extensions for Tobii Pro 3.2 (3.2.1.39) - InstallShield Wizard

Customer Information

Please enter your information.

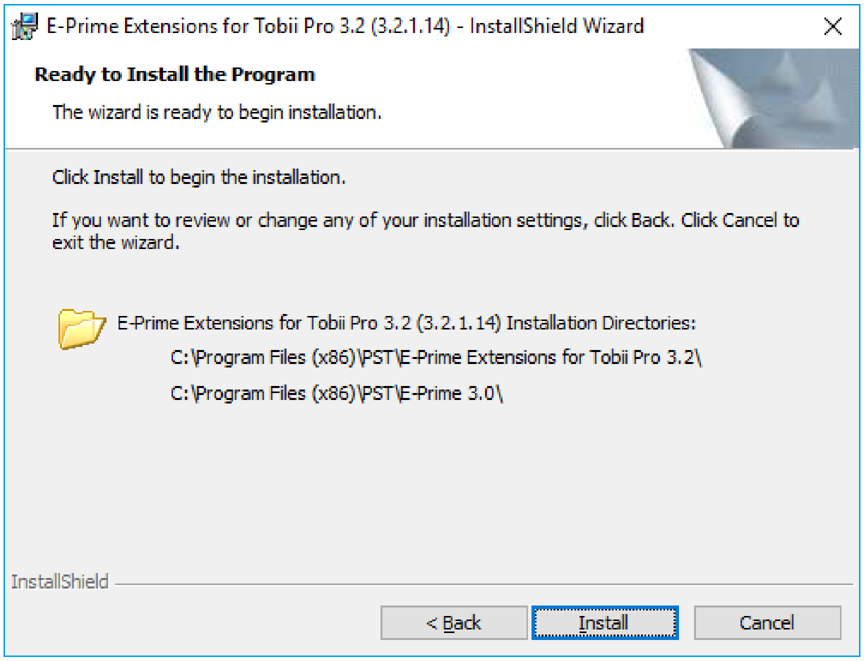
User Name: Windows User

Organization: My Organization

Serial Number: EET 320 -

NOTE: *Please retain the packaging for your EET 3.2 Installation; it contains a label with your serial number which is required for installation.*

4) ***Review*** the installation **folder**. ***Click*** **Install** to continue.



E-Prime Extensions for Tobii Pro 3.2 (3.2.1.14) - InstallShield Wizard

Ready to Install the Program

The wizard is ready to begin installation.

Click Install to begin the installation.

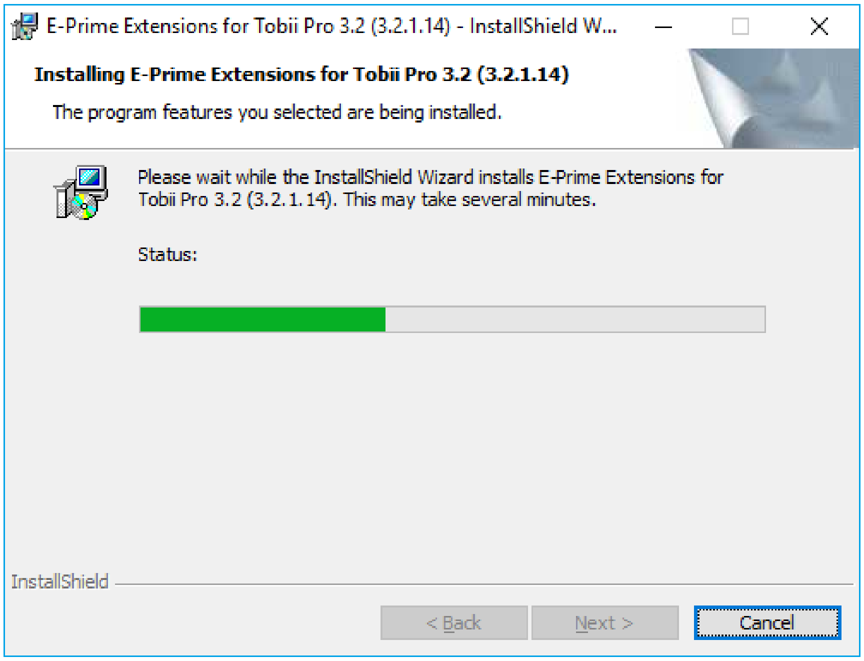
If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

E-Prime Extensions for Tobii Pro 3.2 (3.2.1.14) Installation Directories:

C:\Program Files (x86)\PST\E-Prime Extensions for Tobii Pro 3.2\

C:\Program Files (x86)\PST\E-Prime 3.0\

5) ***Wait*** while the software installs.



E-Prime Extensions for Tobii Pro 3.2 (3.2.1.14) - InstallShield W...

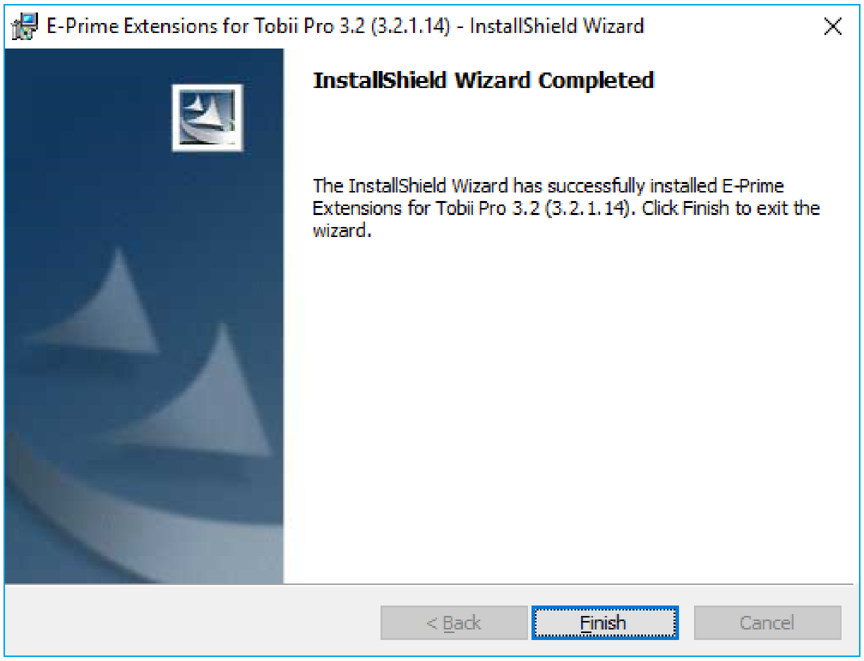
Installing E-Prime Extensions for Tobii Pro 3.2 (3.2.1.14)

The program features you selected are being installed.

Please wait while the InstallShield Wizard installs E-Prime Extensions for Tobii Pro 3.2 (3.2.1.14). This may take several minutes.

Status:

6) ***Wait*** for the confirmation message that the software installs successfully. ***Click*** **Finish** to complete the installation process.



E-Prime Extensions for Tobii Pro 3.2 (3.2.1.14) - InstallShield Wizard

InstallShield Wizard Completed

The InstallShield Wizard has successfully installed E-Prime Extensions for Tobii Pro 3.2 (3.2.1.14). Click Finish to exit the wizard.

2.3. After Installation

Finding the My Experiment Folder

The "…**My Experiments**" folder is frequently used and is the default location to store new experiments created with E-Studio. E-Prime creates the "My Experiments" folder in your personal documents folder on your PC. This folder also contains the "**Samples**" and "**Tutorials**" folders which stores the experiments discussed in *Tutorial 1: Adding Tobii Pro Support to an E-Prime Experiment*.

While the E-Prime documentation directs you to the "…My Experiments" folder, it does not include the full path to the folder. Instead, the documentation refers to "…My Experiments", where the "…" indicates the full path up to your personal documents folder. When you see this notation in the documentation, replace the "…" with the path to your personal documents folder.

NOTE: *The path on your machine may have been modified by your administrator.*

The table below shows the default paths to your personal documents folder:

Folder	Path to your personal documents folder ("My Documents" or "Documents")
*Windows 10	<drive>\Users\<user name>\Documents\My Experiments\3.0\

**Windows 7 and 8 go to "Documents", but it is visible as "My Documents".*

E-Prime Extension for Tobii Pro Samples and Tutorials

The Tobii Pro folder is located within the "…My Experiments" folder, as shown below. The Tobii Pro folder contains the Samples and Tutorials subfolders:

Folder Type	Path to Samples or Tutorials
Samples	…My Experiments\3.0\Tobii Pro\Samples
Tutorials	…My Experiments\3.0\Tobii Pro\Tutorials

NOTE: *For a listing of the installed Samples and Tutorials, please refer to Appendix A: Samples and Tutorials.*

Opening E-Studio

After E-Prime Extensions for Tobii Pro 3.2 installs, launch E-Studio. When E-Studio launches, it detects that the E-Prime Extensions for Tobii Pro 3.2 sample and tutorials files need to be updated. Specifying 'Yes,' allows E-Prime to copy the new Samples and Tutorials into the …My Experiments\3.0 folder.

NOTE: *Copying the Samples and Tutorials upon launching E-Studio overwrites existing Samples and Tutorials. To overwrite the files at a later point in time, refer to Troubleshooting, 4.1 Cannot locate E-Prime Samples and Tutorials.*

Running a Test Experiment

A test experiment should be run after installation to ensure proper functioning of E-Prime Extensions for Tobii 3.2 and the Tobii system. The following test of the EET installation assumes the proper functioning of the Tobii Pro Eye Tracker. If new to Tobii Pro, confirm proper functioning by running and analyzing data from at least one Tobii Pro sample paradigms.

NOTE: *If using the Tobii Pro Lab compatibility features in EET 3.2, we recommend running a test experiment with a Tobii Pro Lab-enabled sample rather than the TET-enabled experiment. This ensures E-Prime is properly configured to access Tobii Pro Lab and that any AOI's, tags, and display events that are created in the experiment are sent to Tobii Pro Lab. The Tobii Pro Lab Project needs to be configured prior to running the experiment in Step 7 (refer to Task 14: Prepare the Tobii Pro Lab Project).*

NOTE: *An EPU (External Processing Unit) should be used with all EPU-enabled eye trackers.*

1) ***Launch*** **E-Studio 3.0**.
2) ***Allow*** E-Prime 3.0 to **copy** the new Samples and Tutorials into your **…My Experiments** folder.
3) ***Navigate*** to **…My Experiments\3.0\Tobii Pro\Samples.**
4) ***Navigate*** to the **TET** folder to run a **TET-enabled** experiment (e.g., **TETFixedPositionAOI.es3**). ***Navigate*** to the **TPL** folder to run a **Tobii Pro Lab-enabled** experiment (e.g., **TPLFixedPositionAOI.es3**).
5) ***Open*** the experiment file.
6) **Check** that the **eye tracker** is **plugged in** and **recognized by your machine.**
7) ***Run*** the **sample experiment**:
   - The track status window will be displayed for you to find your eyes. Once both eyes are shown in the box, press any key to move on to calibration. Follow the dots to complete calibration. When the calibration results are suitable, accept the results to proceed with the experiment.
   - Follow the instructions on screen to perform the task.
   - When the fixation cross (+) is shown, focus on the fixation continuously for two seconds to begin each trial. Alternately, any key may be pressed to continue.

3. How to Create Your Own Tobii Pro Experiment

3.1. Overview

This chapter illustrates the creation of an E-Prime/Tobii Pro experiment using the installed tutorial experiments provided with EET. Additionally, it introduces and explains some programming methods that E-Prime uses.

To complete the tutorials, you need a computer with E-Prime and the EET software already installed. If you have not installed E-Prime 3.0, please complete the installation before continuing. If E-Prime Extensions for Tobii Pro 3.2 is not already installed, please refer to the installation section in this manual (see Chapter 2: *Installing E-Prime Extensions for Tobii Pro*).

While this document includes a basic introduction to programming concepts specific to using E-Prime with Tobii Pro, the tutorial assumes you are familiar with using E-Prime to build behavioral experiments. If you are new to using E-Prime, we recommend that you work through all the tutorials provided on our Product Service and Support website prior to beginning the EET tutorials.

Article 23282: Experiment Design (Beginner)
Article 22741: Experiment Design (Advanced)

3.2. Experiment Design

Overview

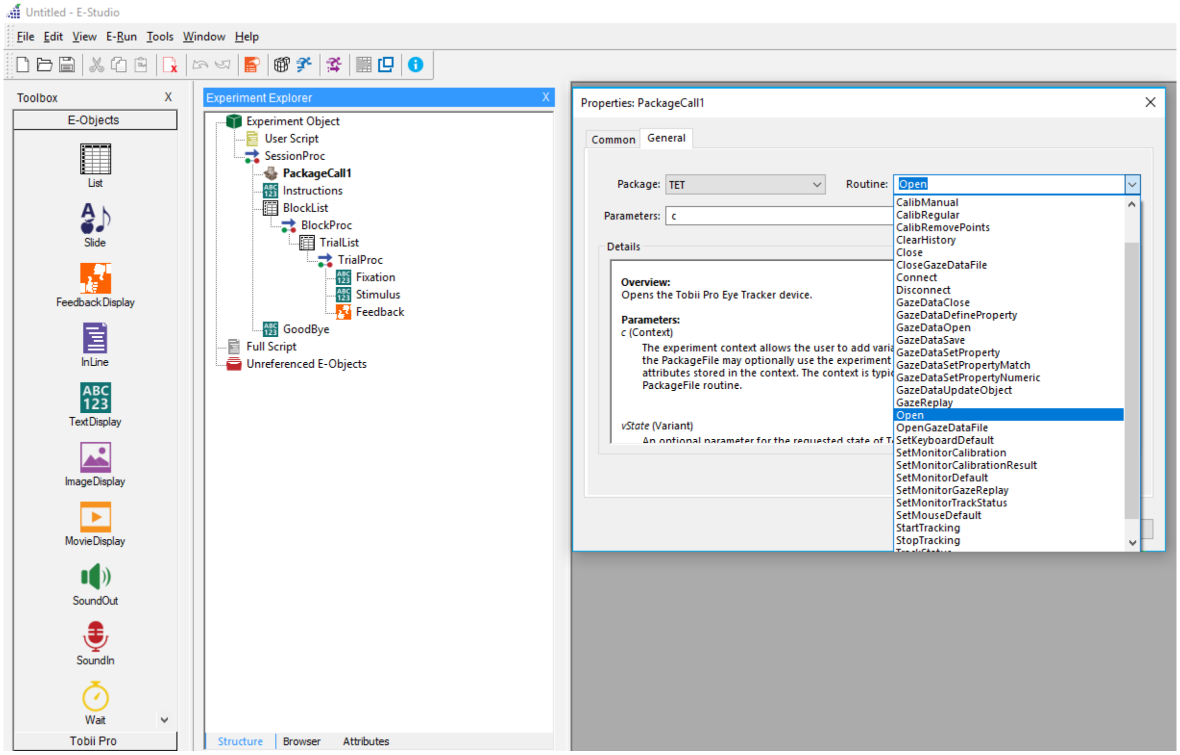
To add support for Tobii Pro to an existing E-Prime experiment, you need the PackageFile feature of E-Prime. An E-Prime **PackageFile** is a pre-written, cohesive set of E-Basic script routines that are grouped together and distributed in a single file. The PackageFile can be authored and maintained externally and then shared amongst different E-Prime experiments. The routines contained in the PackageFile serve to expand the default functionality offered by E-Prime for building experiments that interact with third party hardware and software products.

The E-Prime Extensions for Tobii Pro 3.2 PackageFile is included in the E-Prime Extensions for Tobii Pro installation. The Routines in the PackageFile allow users to automatically perform various functions (e.g., hardware initialization, starting/stopping eye tracker recording, and configuring/sending event markers). These routines are invoked via graphical representations in the Experiment Explorer window in E-Studio. Some of the routines require setting or modifying parameters to enable them to work within the context of an experiment. Once a PackageFile is added to an experiment, there are several methods that can be used to access the routines. A brief overview of each method is provided below.

PackageCall Method

Routines contained within a PackageFile are typically called by dragging a **PackageCall** object from the E-Studio Toolbox. The PackageCall is then dropped into a Procedure Object at the location within the experiment where the call is to be invoked. Once a PackageCall is added to the experiment, you can double click on the object to configure its properties. Minimally, you must specify which Package and routine needs to be called and what parameters needs to be sent during the call.

The PackageCall Properties includes a **Details** section which provides information regarding the purpose of the Routine. A description of any required or optional parameters that need specified to properly use the Routine are also listed. When viewing the Details section, it is helpful to note that a **Remarks** section exists at the bottom of the window. The Remarks section offers further insight on how the PackageCall is used in the experiment.



Package: TET

Routine: Open

Parameters: c

Details

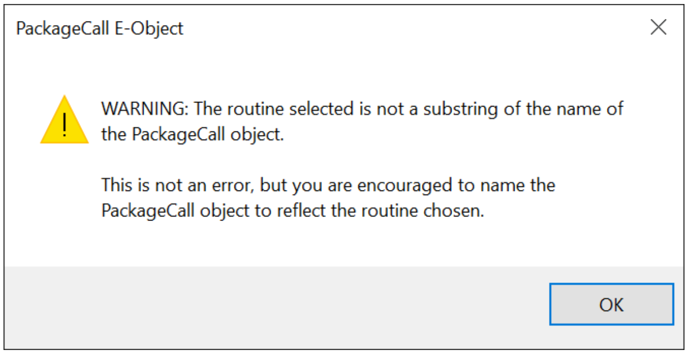
Overview:
Opens the Tobii Pro Eye Tracker device.

Parameters:
c (Context)
The experiment context allows the user to add varia
the PackageFile may optionally use the experiment
attributes stored in the context. The context is typi
PackageFile routine.

vState (Variant)
An optional parameter for the requested state of T

NOTE: *In some cases, you may have to scroll down to view Remarks/additional parameters.*

Once the Package and Routine properties of the PackageCall are set and the Properties dialog is closed, the icon associated with the PackageCall Object changes from the default icon to an author specified icon. If there is no author specified icon associated, then the default icon for the entire PackageFile is used. When dragging and manually configuring a PackageCall, it is strongly recommended that you rename the object to reflect the specific PackageFile and Routines which are being referenced within the object. If you do not follow this recommendation, E-Prime issues the following warning:



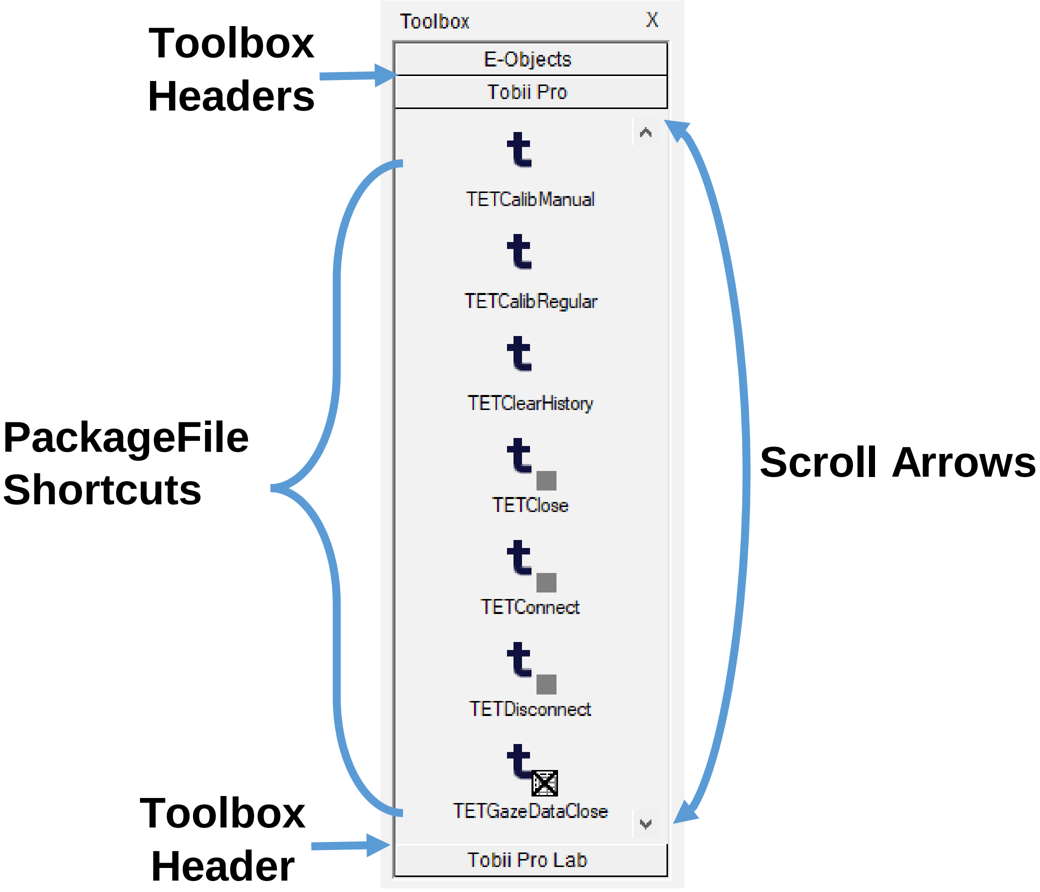
WARNING: The routine selected is not a substring of the name of the PackageCall object.

This is not an error, but you are encouraged to name the PackageCall object to reflect the routine chosen.

It is a common convention to create a name for each object by concatenating together the name of the PackageFile and the name of the Routine being called. For example, if you are using the “TET” PackageFile and calling the “Open” routine, the PackageCall object would be renamed “TETOpen”.

PackageFile Toolbox Shortcuts

A PackageFile author can optionally choose to provide a customized Toolbox of PackageFile Shortcuts to make it faster and easier for the experiment author to access and configure the Routines supported by the PackageFile. If a custom Toolbox is included with the PackageFile, a new panel is added to the Toolbox in E-Studio after the PackageFile is added to the experiment. Each panel of the Toolbox includes a clickable header that indicates which PackageFile is associated with the panel. You can click on the header to show the contents of the Toolbox and then drag a Shortcut from the Toolbox and drop it in a Procedure where the call is to be invoked. Scroll arrows are present on the Toolbox for scrolling through the PackageFile Shortcuts.



Each PackageFile Shortcut contained in the Toolbox is a PackageCall object (as outlined above) which has been predefined by the PackageFile author to save the experimenter time in renaming each PackageCall and configuring its required parameters. It is important to note that PackageFile Shortcuts that have different names in the Toolbox may be configured to reference the same PackageFile routine (e.g., properties may only differ by the list of default parameters that are specified). Providing multiple PackageFile Shortcuts for the same routine can be useful when a single common PackageFile routine is used in different scenarios with the experiment. Using PackageFile Shortcuts are the primary programming method used in the following tutorials.

E-Prime Extensions for Tobii Pro 3.2 provides two different PackageFiles that contain Toolbox Shortcuts. Adding the TET PackageFile enables the Tobii Pro Toolbox Shortcuts. Adding the TobiiProLab PackageFile enables the Tobii Pro Lab Toolbox Shortcuts. The TET and TobiiProLab

PackageFiles can be added simultaneously so researchers can add Shortcuts from both Packages to an experiment. We recommend performing the tutorials in this manual to understand the differences between the two PackageFiles.

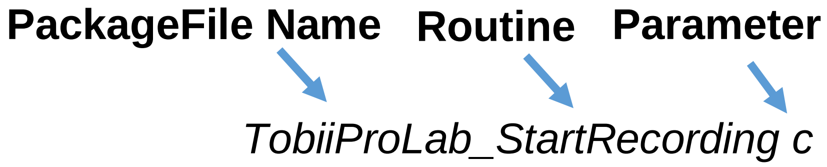
InLine Script Method

An **InLine** is an object used to insert user-defined script into an experiment at a specific point. The object is placed at the desired location in the script. The InLine method can be used to call PackageCalls as well. However, unless there is a need to make a series of PackageCalls in sequence or mingle PackageCalls with additional E-Basic script, we do not typically recommend this calling method.

Calling a PackageFile routine from script is generally the same as calling any other E-Basic Subroutine or Function that could be defined in User Script except that the definition of the routine is contained with the PackageFile rather than within the UserScript section of your experiment.

Due to a common naming convention used by PackageFile authors, a Package routine can be invoked directly from an InLine Object. This is done by concatenating the name of the Package followed by an underscore ("_") and then the name of the Routine you want to call. Lastly, any parameters are added that are required for the call.

For example, the TobiiProLab PackageFile includes a call to start recording in Tobii Pro Lab. The routine is named "StartRecording" and takes the experiment context ("c") as a required parameter. To call this Routine from an InLine Script Object, specify the following in the InLine Object:



TobiiProLab_StartRecording c

To see examples of PackageCall parameters and functions invoked by calling the Routine from an InLine, please refer to:

[Article 26208: Using PackageCalls via InLine Script]

For specific information on PackageCalls in E-Prime Extensions for Tobii Pro 3.2, please refer to:

[E-Prime Command Reference]

3.3. Tutorial 1: Adding Tobii Pro Support to an E-Prime Experiment

Summary:

Incorporating Tobii Pro Support into an existing E-Prime experiment primarily involves adding both the TET PackageFile and the TET PackageCalls to the experimental structure at the appropriate locations. The required parameters may need to be edited to allow the TET Package to meet an experiment's needs.

During this tutorial, E-Prime Extensions for Tobii Pro 3.2 support is added to the FixedPositionAOI.es3 tutorial experiment installed with EET (i.e., …My Experiments\3.0\Tobii Pro\Tutorials\TET). The FixedPositionAOI.es3 experiment consists of one block. Once modified with the Tobii Pro PackageCalls, the beginning of the experiment presents a black box called the TrackStatus window. This allows participants to see if their eyes are captured by the eye tracker. Then, a prompt occurs to calibrate the eye tracker. The calibration is similar to the one that occurs in the Tobii Pro software.

During the stimulus presentation trials, the participant views a Fixation cross and then listens to an animal noise. Next, a picture is shown, and the participant is asked to decide about the relationship of the sound to the picture. Response feedback is provided to the user at the end of each trial.

NOTE*: If not familiar with the* ***FixedPositionAOI.es3*** *experiment, we recommend running the experiment now.*

NOTE: *Tutorials 2 and 3 build upon the concepts learned in this tutorial. They cannot be started until this tutorial has been completed.*

***Goal**:*

This tutorial illustrates how to add the TET PackageCalls into the FixedPositionAOI.es3 tutorial experiment included with E-Prime Extensions for Tobii Pro 3.2. A basic "TET-enabled" paradigm is created when you have completed this tutorial.

Overview of Tasks:

- Open and save FixedPositionAOI.es3.
- Add the TET PackageFile to the Experiment Object.
- Add the TobiiEyeTracker to the Devices tab.
- Edit the TobiiEyeTracker device.
- Add the TET PackageCall to open TET communication.
- Add the TET PackageCall to perform the eye tracker calibration.
- Add the TET PackageCall to close TET communication.
- Add the TET PackageCall to start tracking.
- Add the TET PackageCall to wait until the Fixation Object is fixated upon.
- Modify the Fixation Object.
- Add the TET PackageCall to stop tracking.
- Add the TET PackageCall to replay the gaze data.
- Verify the overall experiment structure and run the experiment.

Recommended readings:
Before beginning the tutorial, we recommend reading and working through the other tutorials provided with E-Prime. Please refer to the following for other available tutorials:

Article 23282: Experiment Design (Beginner)
Article 22741: Experiment Design (Advanced)

Additionally, it is important to understand critical timing in E-Prime to run experiments under the best possible conditions for accurate timing data. Refer to the following articles on timing:

Article 27520: E-Prime Timing Test Results
Article 22852: Timing of E-Objects
Article 22857: Stimulus Preparation Solutions

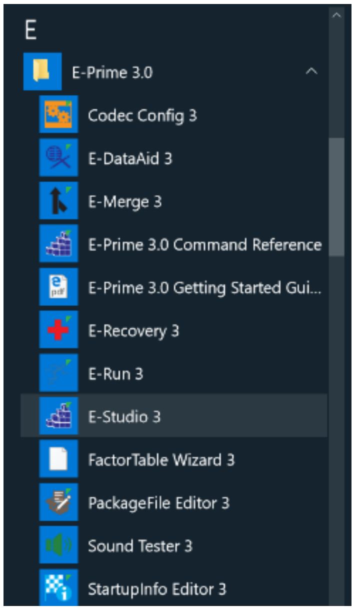
Estimated Tutorial Time:
15-20 minutes

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

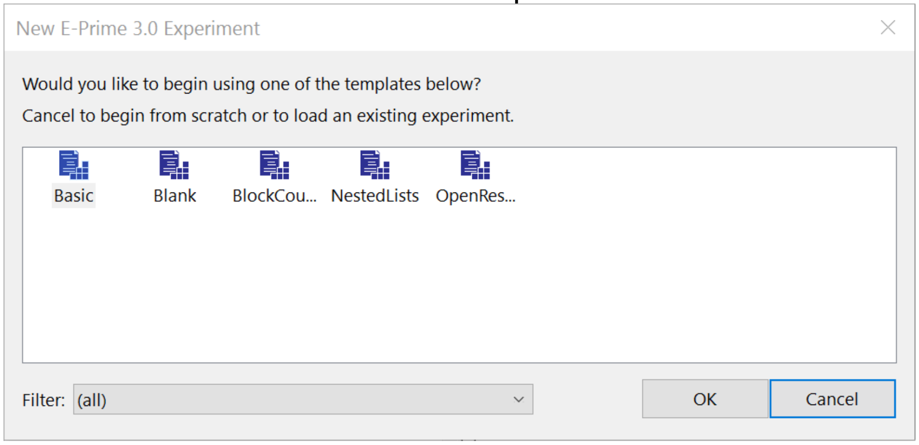
Locate the E-Studio icon in the Start > All Programs > E-Prime 3.0 menu and launch the application by selecting it. Load the FixedPositionAOI.es3 tutorial experiment.

The E-Studio application is installed as part of the typical E-Prime installation. This application is used to create, modify, and test experiments within E-Prime. Open the E-Studio application, navigate to …My Experiments\3.0\Tobii Pro\Tutorials\TET\TETFixedPositionAOI, and load the FixedPositionAOI.es3 tutorial experiment.

1) ***Click*** on the **Windows Start** menu and ***search*** for **E-Prime 3.0**. From the menu, ***click*** on **E-Studio** to launch the application.



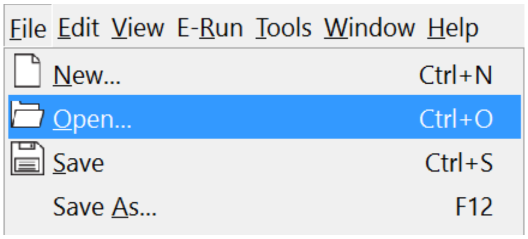
2) ***Click*** the **Cancel** button on the New E-Prime 3.0 Experiment window.



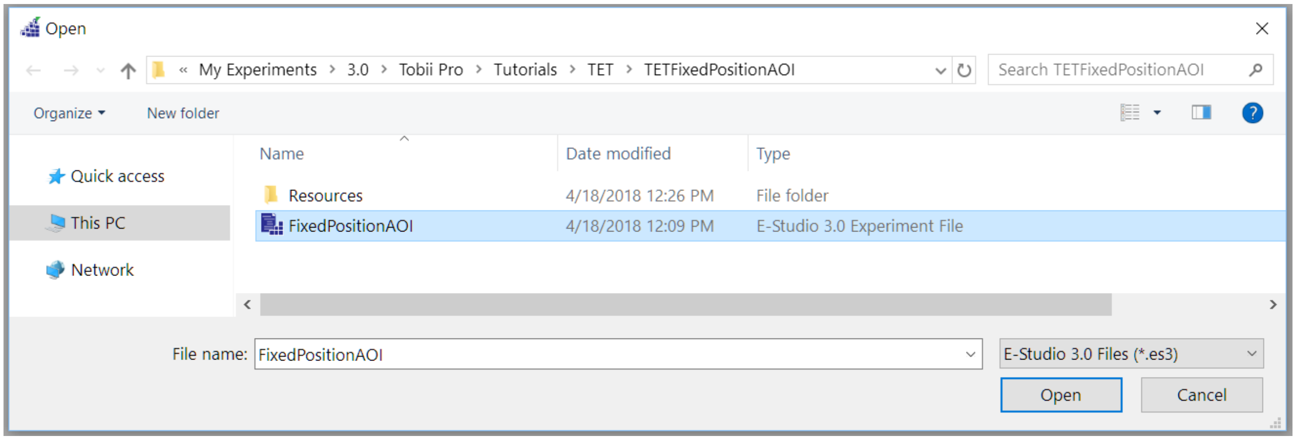
Would you like to begin using one of the templates below?
Cancel to begin from scratch or to load an existing experiment.

Filter: (all)

3) ***Select*** **Open** from the **File** menu**.** ***Navigate*** to the **...My Experiments\3.0\Tobii Pro\Tutorials\TET\TETFixedPositionAOI** folder to load the experiment.



4) ***Select*** the **“FixedPositionAOI.es3”** file and then ***click*** the **Open** button to load the experiment into E-Studio.


Open
« My Experiments › 3.0 › Tobii Pro › Tutorials › TET › TETFixedPositionAOI
Search TETFixedPositionAOI
Organize
New folder
Quick access
This PC
Network
Open
Cancel

Name	Date modified	Type
Resources	4/18/2018 12:26 PM	File folder
FixedPositionAOI	4/18/2018 12:09 PM	E-Studio 3.0 Experiment File

File name: FixedPositionAOI

E-Studio 3.0 Files (*.es3)

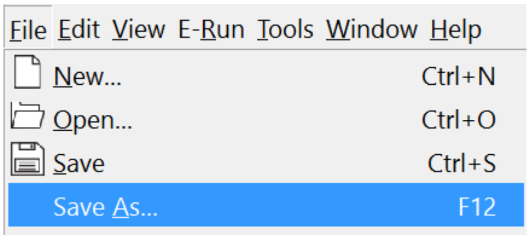
NOTE: *If Samples and Tutorials for E-Prime Extensions for Tobii Pro 3.2 do not exist, see Troubleshooting, 4.1 Cannot locate E-Prime Samples and Tutorials.*

Task 2: Save the experiment under a new name

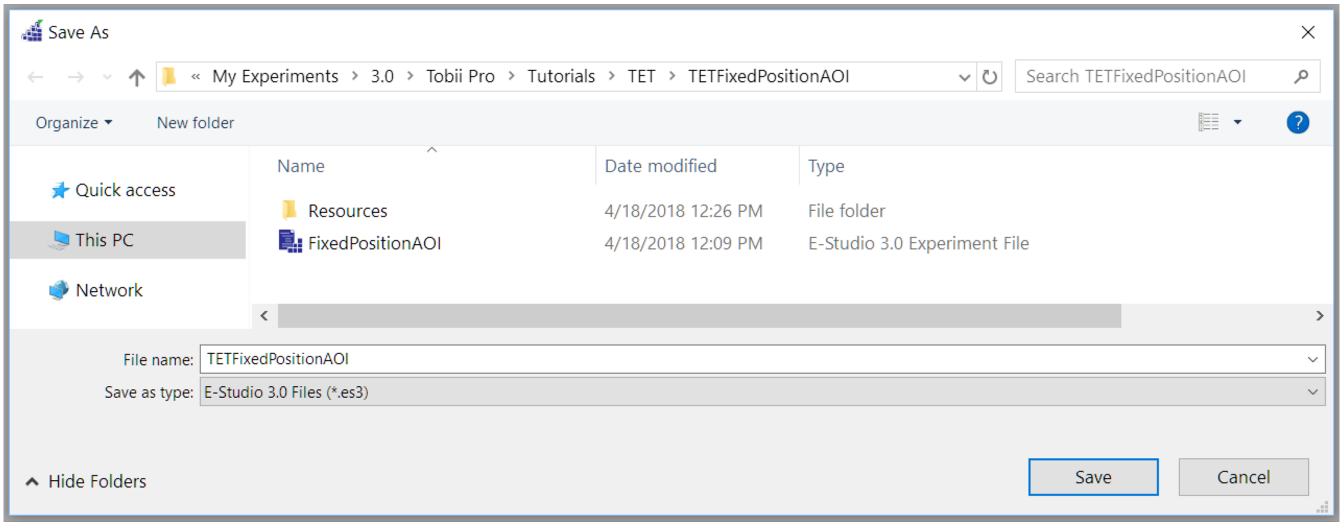
Save the FixedPositionAOI.es3 experiment in the same folder under a new name.

Rename the experiment but be sure to save it in the same folder (…My Experiments\3.0\Tobii Pro\ Tutorials\TET\TETFixedPositionAOI) so that any resources within the experiment will remain valid and can be reused.

1) ***Select*** **Save As...** from the **File** menu**.**



2) ***Type*** "**TETFixedPositionAOI.es3**" as the new name in the **File name** field. ***Click*** the **Save** button.


Save As
« My Experiments › 3.0 › Tobii Pro › Tutorials › TET › TETFixedPositionAOI
Search TETFixedPositionAOI
Organize ▾
New folder
Quick access
This PC
Network
Hide Folders
Save
Cancel

Name	Date modified	Type
Resources	4/18/2018 12:26 PM	File folder
FixedPositionAOI	4/18/2018 12:09 PM	E-Studio 3.0 Experiment File

File name: TETFixedPositionAOI

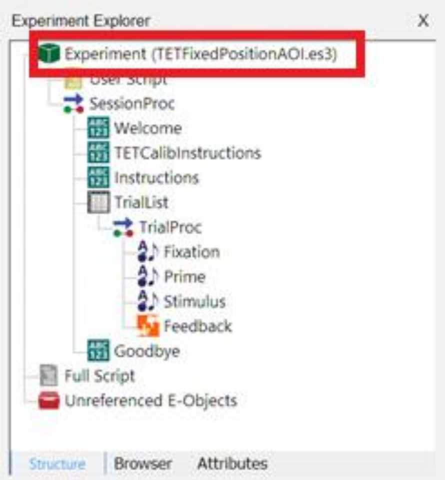
Save as type: E-Studio 3.0 Files (*.es3)

Task 3: Add the TET Package to the Experiment Property Pages

Open the Properties dialog for the Experiment Object and use the Packages tab to add the TET PackageFile to the experiment.

PackageFiles in E-Prime are cohesive sets of E-Basic routines that are grouped together. To gain access to the routines within a PackageFile, the PackageFile needs added to the experiment. PackageFiles can be added to an experiment using the Packages tab of the Experiment Object properties. The routines that are used to communicate with the Tobii Pro hardware/software at runtime are contained within the TET PackageFile. The PackageFile also includes a customized Toolbox which contains shortcuts that can be dragged into Procedures within an experiment to invoke routines contained in the PackageFile.

1) ***Double click*** the **Experiment Object** at the top of the tree in the **Experiment Explorer** window.



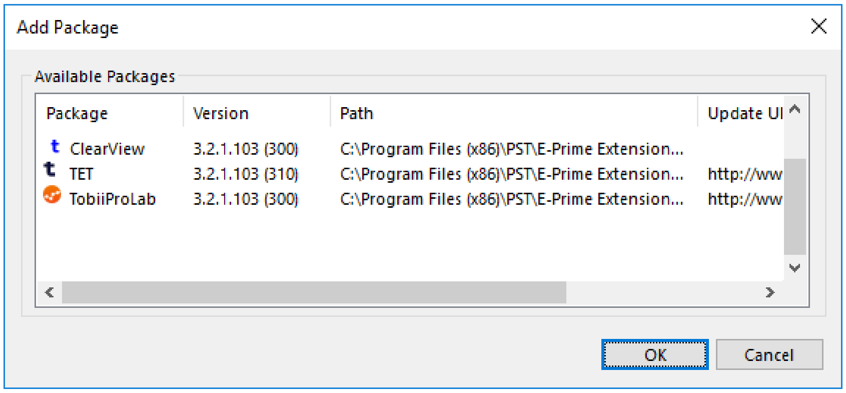
Experiment Explorer

```
Experiment (TETFixedPositionAOI.es3)
    User Script
    SessionProc
        Welcome
        TETCalibInstructions
        Instructions
        TrialList
            TrialProc
                Fixation
                Prime
                Stimulus
                Feedback
        Goodbye
Full Script
Unreferenced E-Objects
```

2) ***Click*** on the **Packages** tab of the **Experiment Object Properties** dialog.

3) ***Click*** the **Add...** button.

4) ***Select*** the **TET PackageFile** in the **Add Package** dialog.



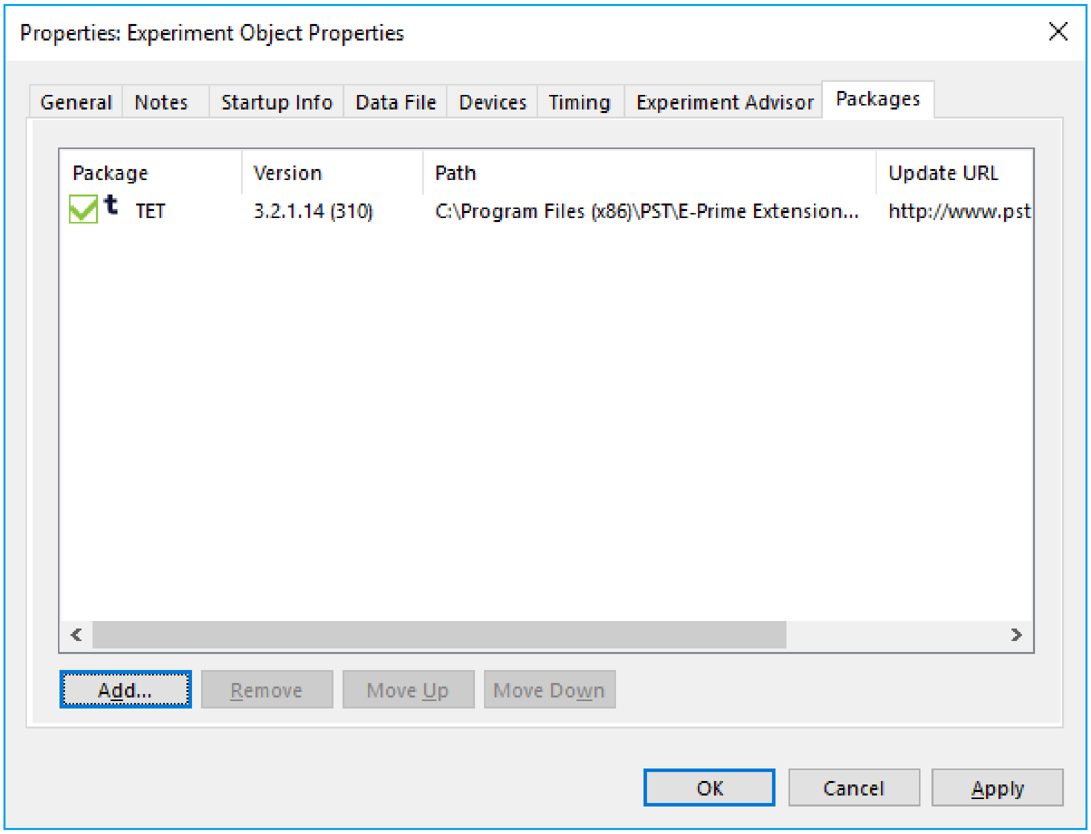
Add Package

Available Packages

Package	Version	Path	Update UI
ClearView	3.2.1.103 (300)	C:\Program Files (x86)\PST\E-Prime Extension...	
TET	3.2.1.103 (310)	C:\Program Files (x86)\PST\E-Prime Extension...	http://ww
TobiiProLab	3.2.1.103 (300)	C:\Program Files (x86)\PST\E-Prime Extension...	http://ww

5) ***Click*** the **OK** button to accept the changes.

6) ***Verify*** the **TET PackageFile** has been added and is checked (**<u>Do not</u>** yet ***click*** **OK** or ***dismiss*** the dialog).



Package	Version	Path	Update URL
☑ TET	3.2.1.14 (310)	C:\Program Files (x86)\PST\E-Prime Extension...	http://www.pst

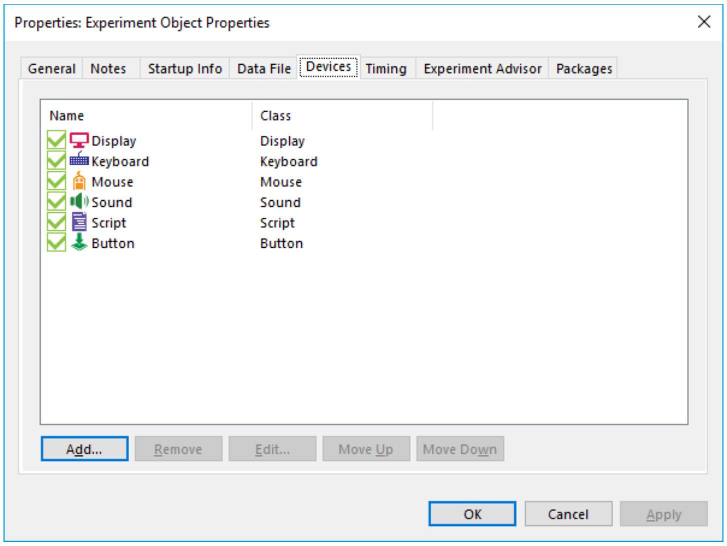
NOTE: *The PackageFile version number that is displayed by E-Studio reflects the version of the TET PackageFile that is currently installed on your machine and may not match the picture.*

Task 4: Add the TobiiEyeTracker Device to the Experiment Properties

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

To run an experiment with Tobii Pro capabilities, the eye tracker needs to be added to the list of available devices in the experiment. Select the Devices tab of the Experiment Object Property Pages to add the TobiiEyeTracker Device to the list of devices. Verify that it is the last device shown.

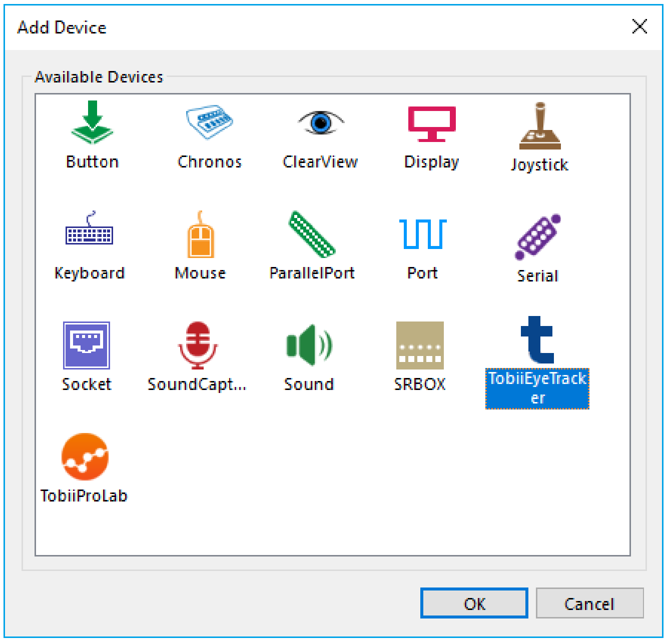
1) ***Click*** on the **Devices** tab of the **Experiment Object Properties** dialog.



Name	Class
☑ Display	Display
☑ Keyboard	Keyboard
☑ Mouse	Mouse
☑ Sound	Sound
☑ Script	Script
☑ Button	Button

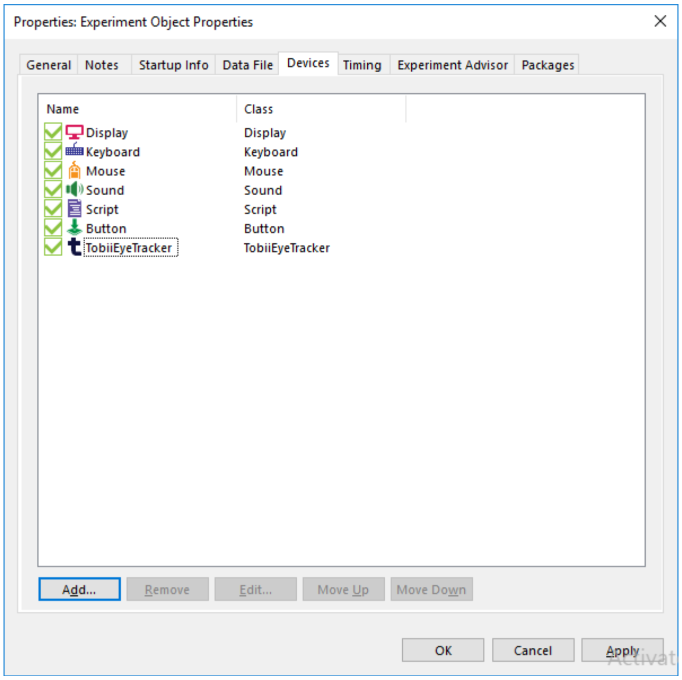
2) ***Click*** the **Add...** button.

3) ***Select*** the **TobiiEyeTracker** Device in the **Add Device** dialog.



4) ***Click*** the **OK** button to accept the changes.

5) ***Verify*** the **TobiiEyeTracker Device** is listed last under the **Name** column and is checked.



Properties: Experiment Object Properties

Name	Class
☑ Display	Display
☑ Keyboard	Keyboard
☑ Mouse	Mouse
☑ Sound	Sound
☑ Script	Script
☑ Button	Button
☑ TobiiEyeTracker	TobiiEyeTracker

NOTE: *The order of devices from top to bottom is the order in which the devices become initialized in the beginning of an experimental run. Do not move the TobiiEyeTracker Device to the top as the Display device must be opened prior to the TobiiEyeTracker Device. If this order is not followed, the experiment terminates.*

Task 5: Edit the TobiiEyeTracker Device Properties

Open and edit the Property Pages for the TobiiEyeTracker Device.

The TobiiEyeTracker Device properties can be configured to meet an eye tracker's or experiment's needs. Before running a TET-enabled experiment, be sure to edit the TobiiEyeTracker Device properties for the eye tracker used during runtime.

The first property (**Name**) allows for customization of the TobiiEyeTracker Device name. The default value for this property is TobiiEyeTracker.

The **Eye Tracker** property allows for selection of the eye tracker to be used during an experiment run. A button is available next to the Eye Tracker property that launches Tobii Pro Eye Tracker Manager program. Please refer to *Tobii Pro Eye Tracker Manager* for more information.

The **Frequency** property allows users to specify a frequency value for the eye tracker. A proper frequency supported by the eye tracker must be selected or else a runtime error occurs.

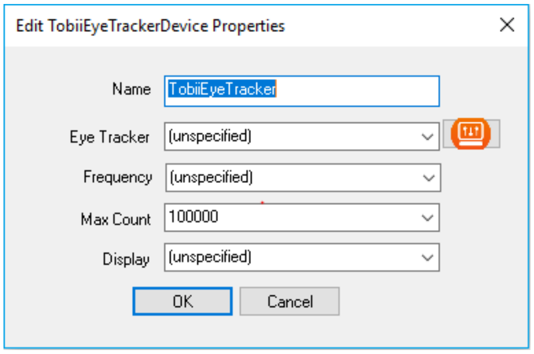
The **Max Count** property sets the gaze data buffer size. This property controls the amount of time that the eye tracker can save eye gaze data before the oldest data in the buffer is overwritten. The default value for this property is 100,000. Refer to *Appendix C: Timing and Synchronization: Writing to the Buffer and Buffer Size* for more information.

The **Display** property allows researchers to switch which active Display device the eye tracker is being used with. Refer to *Tutorial 3: Multiple Monitors: Creating a Participant Station* for more information on changing the Display property.

NOTE: *Eye Tracker device properties with a value of (unspecified) use the value last used from the Tobii Pro Eye Tracker Manager software.*

1) ***Double click*** **TobiiEyeTracker** to display the **TobiiEyeTracker Device Properties** dialog.

2) ***Verify*** the **Name**, **Eye Tacker**, **Frequency**, **Max Count**, and **Display** Properties. ***Specify*** **appropriate values** for the experiment.



Edit TobiiEyeTrackerDevice Properties

Name: TobiiEyeTracker

Eye Tracker: (unspecified)

Frequency: (unspecified)

Max Count: 100000

Display: (unspecified)

3) ***Click*** the **OK** button to accept the **TobiiEyeTracker Device Properties dialog**. ***Click*** the **OK** button again to accept the **Experiment Object Properties dialog**.

Task 6: Add the TETOpen PackageCall to Enable Tobii Pro Eye Tracking

Add a PackageCall to the SessionProc that opens TET communication.

The **TETOpen** PackageCall opens communication to the Tobii Pro eye tracking server in preparation for eye movement data collection. The Open routine confirms that the TobiiEyeTracker Device is open and enables TET PackageCalls to execute. The TETOpen PackageCall is added at the beginning of the SessionProc for this purpose.

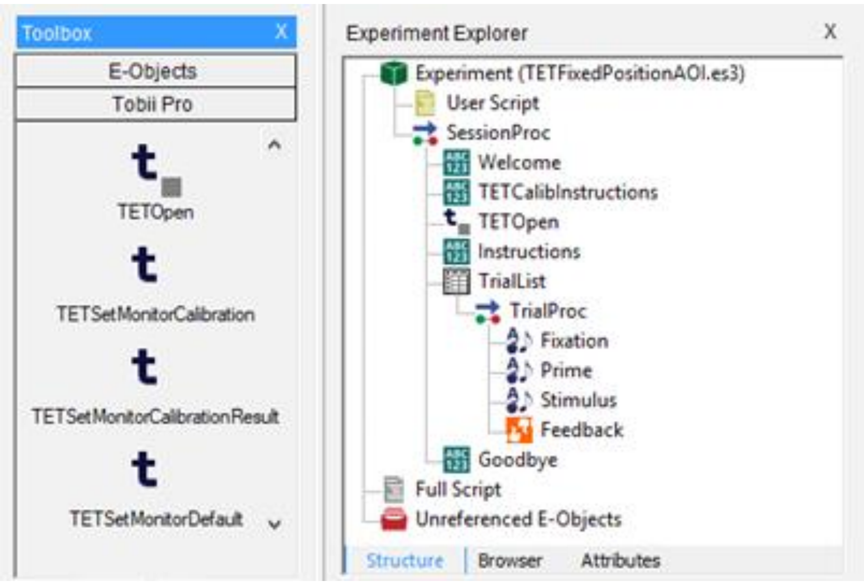
The first parameter (“**c**”) is the common to most PackageCall routines and passes in the current experiment Context. This parameter is the only required parameter for TETOpen.

The second parameter (**vState**) is an optional parameter. This parameter controls the state of the communications between E-Prime and the Tobii Eye Tracker (i.e., “on” or “off”). If not specified, this defaults to "on." For information on how to run an EET-enabled experiment without an eye tracker connected, please refer to *Troubleshooting, 4.2 Running without an eye tracker*.

The third parameter (**vConnect**) is an optional parameter that specifies if a connection should be immediately made to the Tobii Eye Tracker Server. Accepted values are True and False. If not specified, this defaults to True. Setting this value to false is useful whenever the eye tracker does not need to be connected to immediately (i.e., running trials without eye tracking).

The final parameter (**vHandlePreRelease**) is an optional parameter that specifies if the HandlePreRelease routine is called prior to the execution of this routine's script. If not specified, this defaults to True.

1) ***Drag*** the **TETOpen Object** from the **Tobii Pro Toolbox** and ***drop*** it in the **SessionProc** Procedure **after TETCalibInstructions**.



Task 7: Add the TETCalibRegular PackageCall to Enable the Calibration Routines

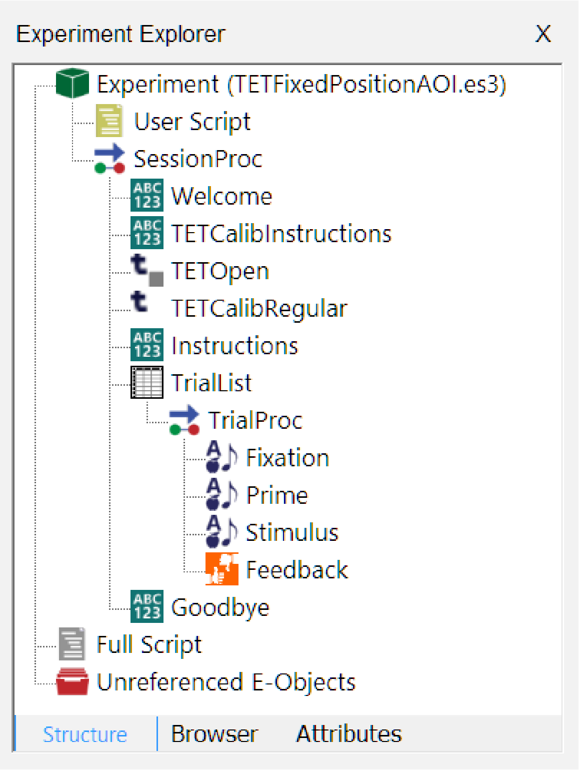
Add a PackageCall to the SessionProc to perform a calibration of the participant's eyes to the eye tracker.

The **TETCalibRegular** PackageCall is used to display the Track Status window and calibrate the participant. Calibration can be started with any key press. At the end of the calibration results are shown of the calibration and a prompt occurs to accept or reject the results.

The TrackStatus window appears before Calibration and shows when the eye tracker detects eyes by displaying two white circles in the box. If eyes are detected, a green box is shown along the bottom of the box that has the word "Both" written in it. If the eye tracker detects only one eye then the TrackStatus box shows only one white circle, the bar is red, and the text indicates which eye the eye tracker detects.

The default number of calibration points is 5 in the PackageCall Toolbox Shortcut. This can be customized by changing the **vCalibNumPointsOverride** parameter to another value. The only values that can be specified are 2, 5, and 9. If a value other than the allowed values is specified, an error occurs.

1) ***Drag*** the **TETCalibRegular** object from the **Tobii Pro Toolbox** and ***drop*** it ***after*** **TETOpen** in the **SessionProc.**



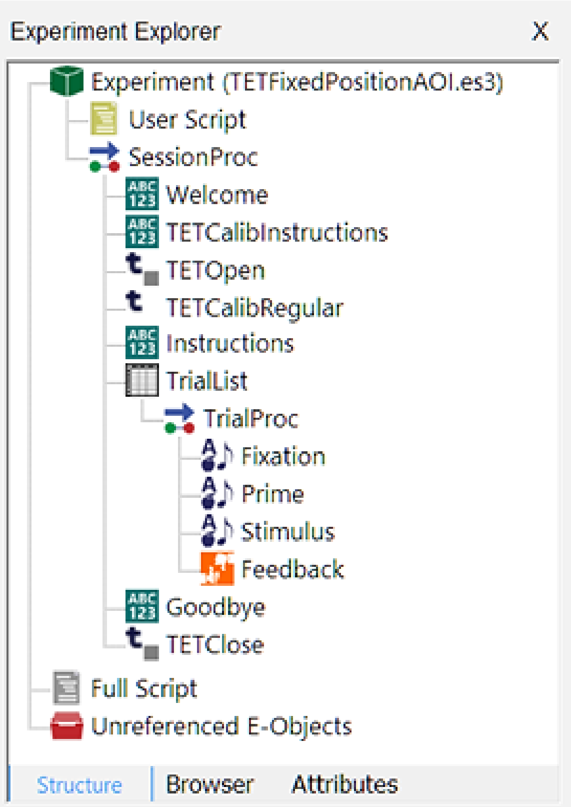
```
Experiment Explorer                                X
Experiment (TETFixedPositionAOI.es3)
    User Script
    SessionProc
        Welcome
        TETCalibInstructions
        TETOpen
        TETCalibRegular
        Instructions
        TrialList
            TrialProc
                Fixation
                Prime
                Stimulus
                Feedback
        Goodbye
Full Script
Unreferenced E-Objects
Structure    Browser    Attributes
```

Task 8: Add the TETClose PackageCall to Disable Tobii Pro Eye Tracking

Add a PackageCall to the SessionProc that closes TET communication.

The last necessary PackageCall in any TET-enabled experiment is **TETClose**. This call closes the TobiiEyeTracker Device, disconnects the TobiiEyeTracker from the server, and stops any eye tracking that may still be happening. It should be called once eye tracking is no longer necessary in the experiment. Therefore, no more TET routines should occur after Close is called.

1) ***Drag*** the **TETClose** object from the **Tobii Pro Toolbox** and ***drop*** it ***after*** the **Goodbye** object in the **SessionProc**.



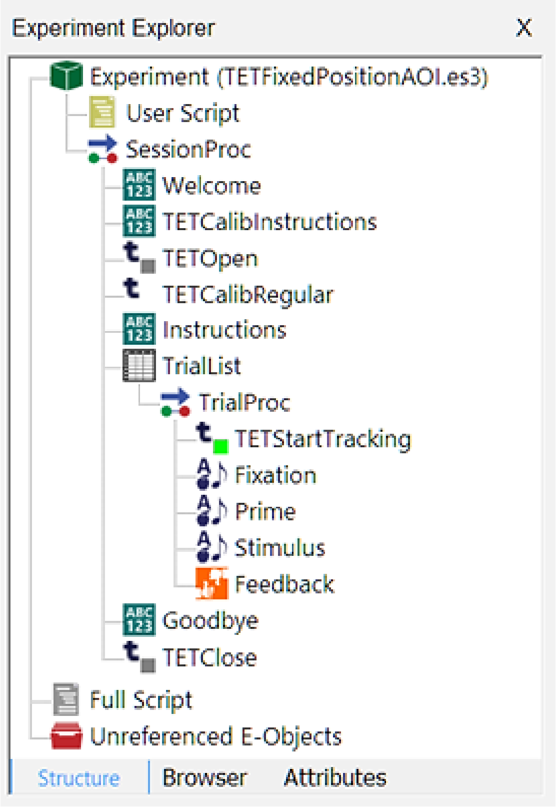
Task 9: Add the TETStartTracking PackageCall to Begin Tracking Eye Movements

Add a PackageCall to the TrialProc to start the eye tracker recording.

The next step is to add the **TETStartTracking** PackageCall. This PackageCall is placed at the location in the procedure where eye tracking movements need to start being tracked. This is often at the beginning of a trial but may vary depending on the experiment.

TETStartTracking offers an optional parameter (**vTrackingMode**) that allows for the tracking mode to be changed. There are 3 modes that can be used for tracking: Latch, Wait, and Immediate. Latch is the default mode. When the tracking mode is Latch or Immediate, the StartTracking functionality is performed and the routine completes. Wait mode starts tracking once the Tobii Pro Eye Tracker history count has increased. For most studies, we recommend using Latch because it is the most synchronized in terms of when the eye tracker is started and receiving gaze data.

1) ***Drag*** the **TETStartTracking** object from the **Tobii Pro Toolbox** and ***drop*** it as the first object in the **TrialProc** procedure.



Task 10: Add the TETWaitForFixation PackageCall to Halt the Trial until the Participant Fixates on an Object for a Given Amount of Time

Add a PackageCall to the TrialProc that requires the participant's fixation before continuing the experiment.

The next step is to add the **TETWaitForFixation** PackageCall. This PackageCall is used to halt the beginning of a trial until the participant fixates on an object on the screen for a set amount of time. In this example, the experiment presents a simple fixation cross (+) at the start of each trial. The PackageCall object should always follow the Slide Object being used to present the fixation. Additionally, the sub-object (contained on the Slide) that presents the visual stimulus used as a fixation must be named “Fixation”.

Just like TETOpen, the first parameter is “**c**.” It passes in the current experiment Context.

The second parameter (**theSlide**) is a required parameter that specifies the name of the Slide object that presents the fixation. By default, the PackageCall Shortcut specifies an object named Fixation, which can be changed if necessary. For example, if the Slide that presents the Fixation is named “SlideStartTrial”, then SlideStartTrial needs to be specified for this parameter. Note that the Slide object presenting the fixation must always contain a sub-object named Fixation that presents the fixation point. The sub-object must have its BorderWidth property set to a value greater than 0 if visual feedback is being used. This is discussed further in *Task 11: Confirm the Sub-Object on the Fixation Slide Object displays the Desired Fixation*.

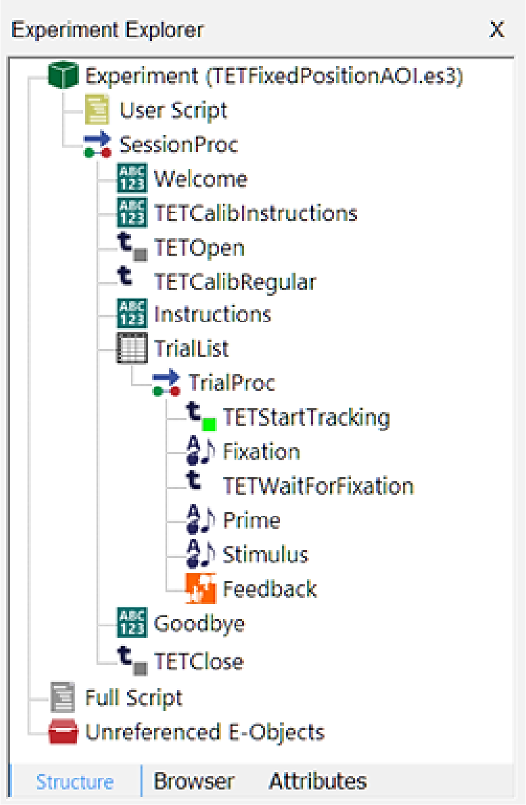
The third parameter (**nMinFixationDuration**) is a required parameter that specifies the amount of time in milliseconds that the participant must maintain fixation before **theSlide** terminates. The default value of the PackageCall Shortcut is set to 2000.

The fourth parameter (**vVisualFeedbackColor**) is an optional parameter that specifies the color of the visual feedback presented to the participant when the appropriate sub-object is fixated upon (e.g., “green” which is the default). This changes the BorderColor of the sub-object named “Fixation” when the participant's gaze is on the fixation.

The fifth parameter (**vKey**) is also optional. This parameter specifies the key that is used to simulate a response when the fixation has occurred. If not specified, the "1" key is used.

The final parameter (**vHandleTracking**) is optional. It specifies whether TETStartTracking and TETStopTracking should occur. This is useful during Tobii Pro Lab-enabled experiments when the start and stop tracking routines are not typically used (refer to *Tutorial 5: Adding Tobii Pro Lab Support to an Experiment*). The default value is “True”. If set to “False”, the Fixation Object may not occur properly.

1) ***Drag*** the **TETWaitForFixation object** from the **Tobii Pro Toolbox** and ***drop*** it ***after*** the **Fixation object** in the **TrialProc** procedure.


Experiment Explorer X

Experiment (TETFixedPositionAOI.es3)
- User Script
- SessionProc
  - Welcome
  - TETCalibInstructions
  - TETOpen
  - TETCalibRegular
  - Instructions
  - TrialList
    - TrialProc
      - TETStartTracking
      - Fixation
      - TETWaitForFixation
      - Prime
      - Stimulus
      - Feedback
  - Goodbye
  - TETClose
- Full Script
- Unreferenced E-Objects

NOTE: *Because the TETWaitForFixation PackageCall Shortcut's default parameters correspond to the values that are appropriate for this experiment, no edits of the parameters are needed. However, this would not be the case if the Fixation object was named differently or if the PackageCall was manually configured (e.g., drag over the PackageCall object and specify the Package and Routine).*

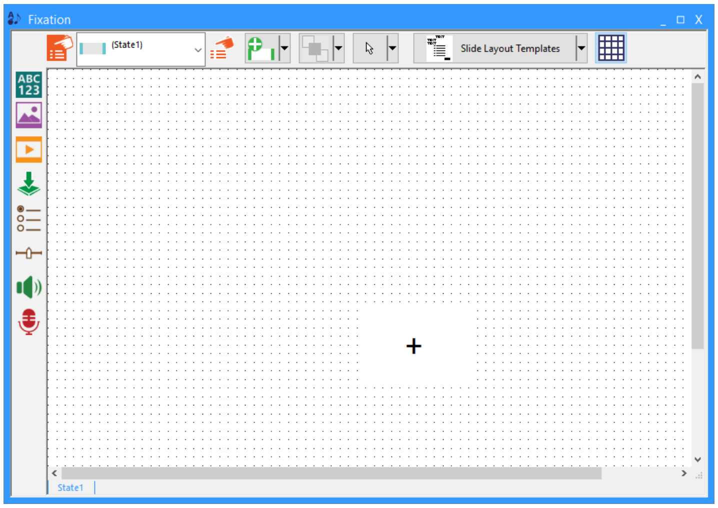
Task 11: Confirm the Sub-Object on the Fixation Slide Object displays the Desired Fixation

Confirm the Fixation Slide has a Fixation sub-object with appropriate text.

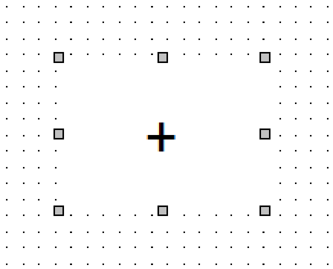
In the last step, we confirmed that the TETWaitForFixation PackageCall interacts with the Slide object named Fixation. Now we need to confirm that the Slide object displays a "+" on a sub-object that is also named "Fixation." This step can be confusing because in this example both the Slide and the Slide sub-object are named "Fixation."

NOTE: *When using the TETWaitForFixation PackageCall, it is imperative to name the Slide sub-object "Fixation." Otherwise the TETWaitForFixation PackageCall does not function properly.*

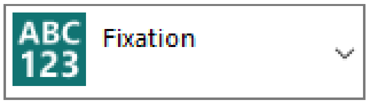
1) ***Double click*** the **Fixation Slide** to open it in the workspace.



2) ***Click*** on the **fixation cross** (+) sub-object.



3) ***View*** the **Sub-Object dropdown menu*****.*** ***Confirm*** the **name** of the sub-object that presents the fixation cross is **"Fixation"**.



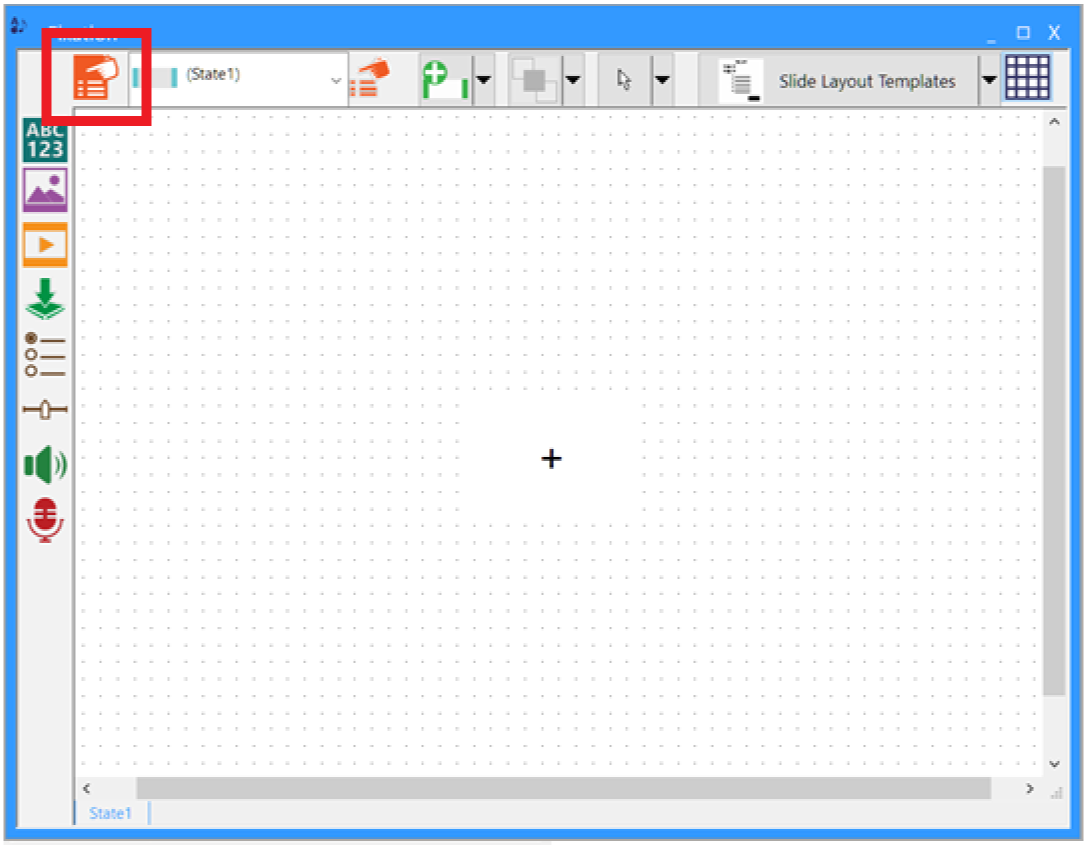
NOTE: *The SlideText sub-object BorderWidth property must be set to a value greater than 0.*

Task 12: Modify the Fixation Slide Object to work in Conjunction with the TETWaitForFixation PackageCall

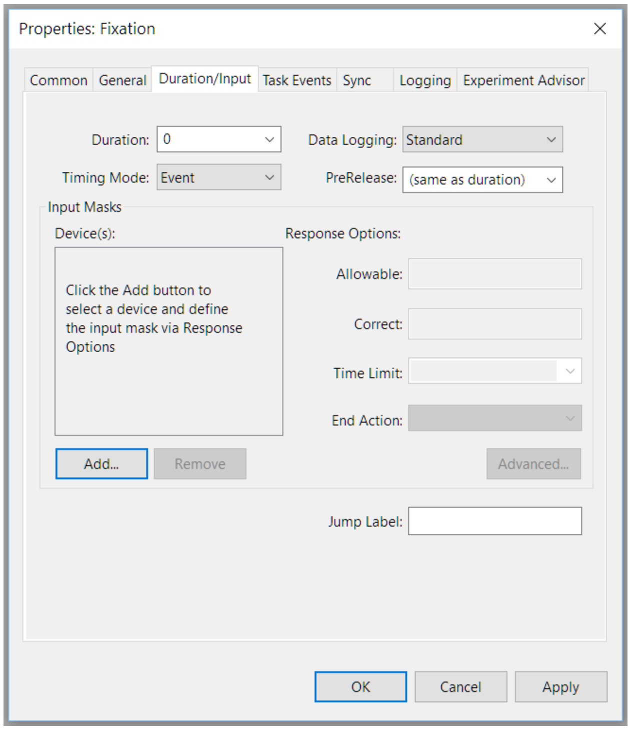
Configure the Fixation object to accept input from the TETWaitForFixation PackageCall.

The general method employed by TETWaitForFixation is one of active processing: the PackageCall keeps track of the participant's fixations by actively accessing the eye tracking data stream and determining the location of the participants gaze. Once a fixation is maintained on the Fixation sub-object for the specified amount of time, the PackageCall terminates the object's execution by simulating a response to the object. Therefore, the object must have an Input Mask, along with specific values for the TimeLimit and EndAction properties that enable the simulated response to terminate the object. For more information on E-Prime Input Masks, see *Article 22723: Input Devices*.

1) ***Click*** the Slide object's **Property Pages** button.



2) ***Select*** the **Duration/Input** tab.



Properties: Fixation

Duration: 0

Data Logging: Standard

Timing Mode: Event

PreRelease: (same as duration)

Input Masks

Device(s):

Click the Add button to select a device and define the input mask via Response Options

Response Options:

Allowable:

Correct:

Time Limit:

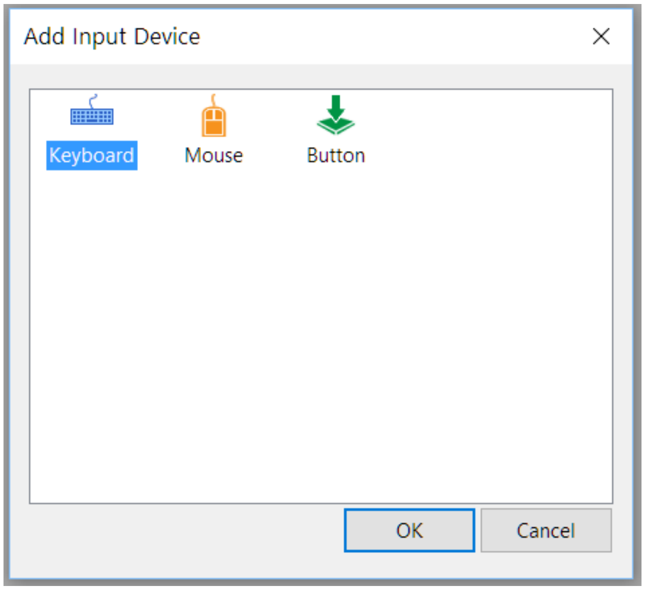
End Action:

Jump Label:

3) ***Edit*** the **Duration** property to equal **0**.

4) ***Confirm*** the **Data Logging** property is set to **Standard**.

5) ***Click*** the **Add** button in the **Input Masks** area to add a Keyboard device.



Add Input Device

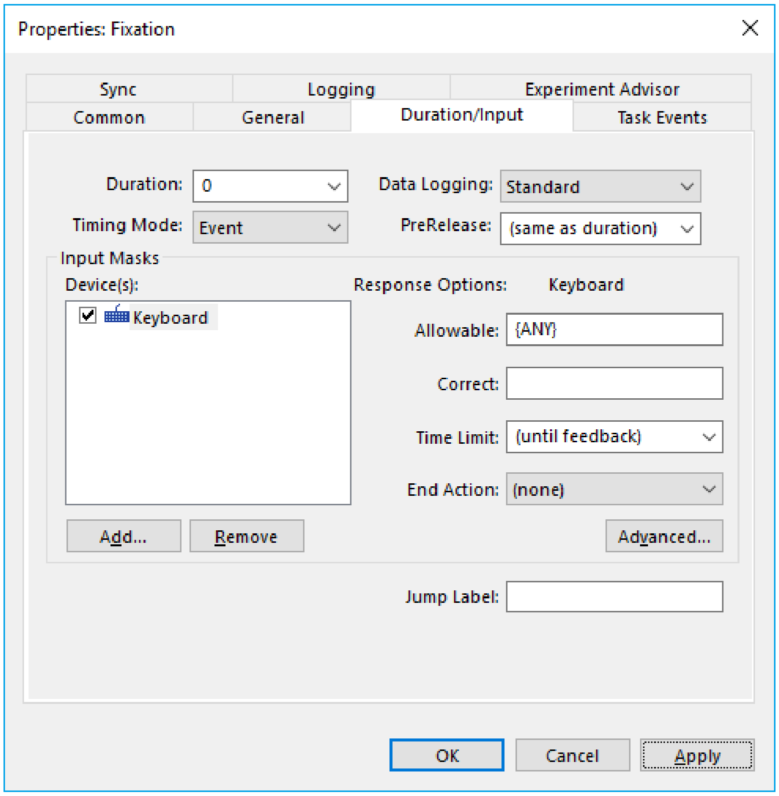
6) ***Select*** **Keyboard** and ***click*** **OK**.

7) ***Verify*** that **Keyboard** is listed under **Device(s)** and that it is checked.

8) ***Verify*** that the **Allowable** property is set to “**{ANY}**”.

9) ***Edit*** the **Time Limit** property to be “**(until feedback)**”.

10) ***Edit*** the **End Action** property to be “**(none)**”.



Properties: Fixation

Duration: 0
Timing Mode: Event
Data Logging: Standard
PreRelease: (same as duration)

Input Masks
Device(s): ☑ Keyboard
Response Options: Keyboard
Allowable: {ANY}
Correct:
Time Limit: (until feedback)
End Action: (none)

Jump Label:

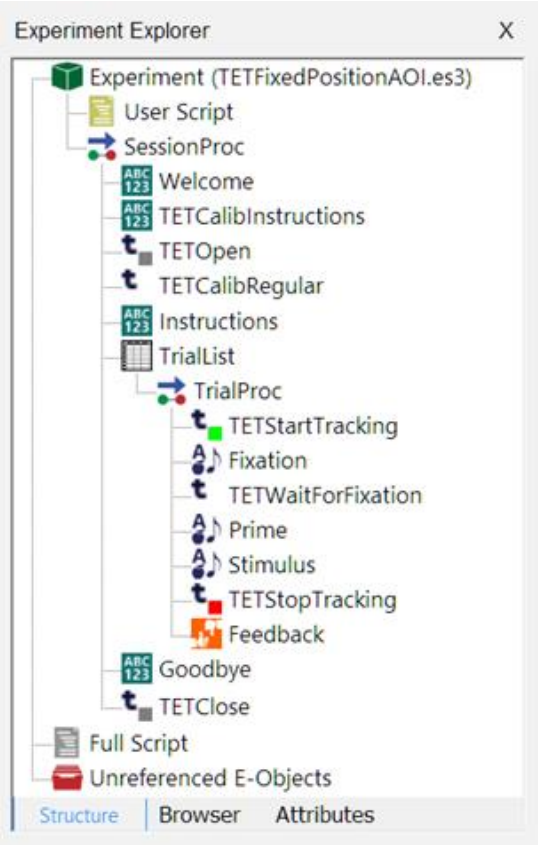
11) ***Click*** the **OK** button to accept the changes.

Task 13: Add the TETStopTracking PackageCall to Stop Tracking Eye Movements

Add a PackageCall to the TrialProc to stop the eye tracking.

Once the critical stimuli have been presented, it is appropriate to call the **TETStopTracking** PackageCall. It is rare that an experimenter wants to collect gaze data on the Feedback object, so a common place for this PackageCall is immediately after the last critical stimulus and prior to the Feedback object.

1) ***Drag*** the **TETStopTracking** object from the **Tobii Pro Toolbox** and ***drop*** it after the **Stimulus** object in the **TrialProc** procedure.



Task 14: Add the TETGazeReplay PackageCall to Visually Replay the Eye Movements

Add a PackageCall to the TrialProc that replays the gaze of the participant.

The PackageCall **TETGazeReplay** visually replays the samples that the Tobii Eye Tracker collected during the stimulus presentation. This can typically be placed in an experiment as the last object in the Procedure that shows critical stimuli. This call is typically used to verify acceptable functioning of the eye tracker during an experiment. However, it is common that gaze data is not replayed to a participant during a study. Typically, the PackageCall is removed prior to data collection.

Just like TETOpen and TETWaitForFixation, the first parameter is “**c**.” It passes in the current experiment Context.

The second parameter (**theSlide**) is a required parameter. This is the name of the Slide object that the GazeReplay replays the eye tracking samples for. For example, if the Slide showing the critical stimuli is named “Stimulus”, then Stimulus needs to be specified for this parameter.

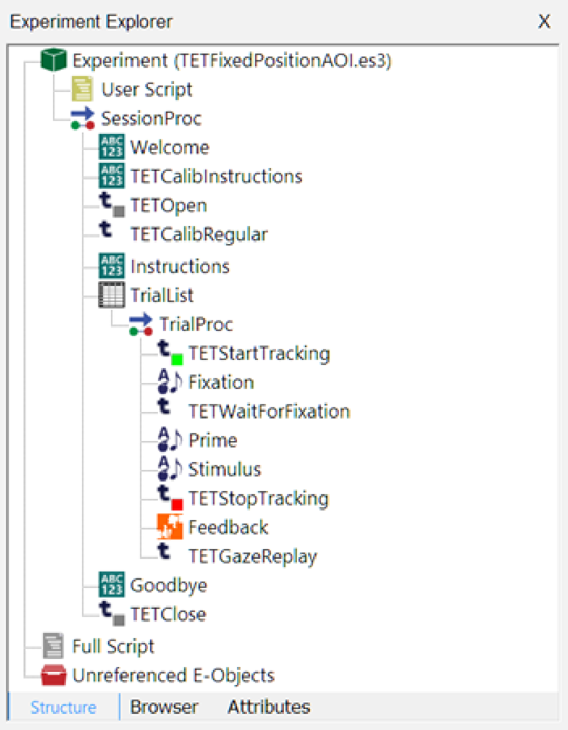
The third parameter (**vStartTime**) is an optional parameter that creates a start time stamp for the first gaze point to be included in the replay. Typically, this is specified as the OnsetTime of an object. If not specified, this defaults to the first sample in the history.

The fourth parameter (**vStopTime**) is an optional parameter that works like vStartTime. However, this creates a time stamp for the last gaze point to be included in the replay. If not specified, this defaults to the last sample in the history.

The fifth parameter (**vHandlePreRelease**) is an optional parameter that specifies if the HandlePreRelease routine is called prior to the execution of this routine’s script. If not specified, this defaults to True.

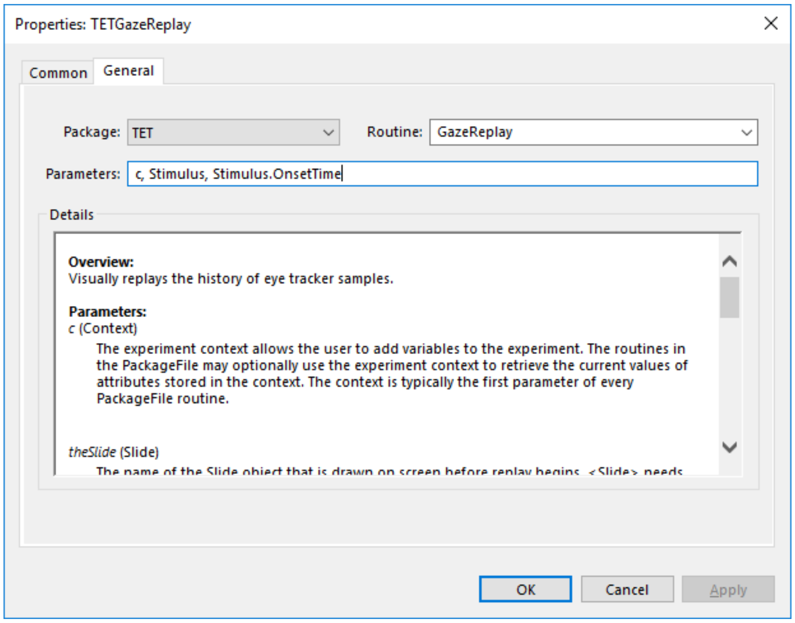
The final parameter (**vPlaybackRate**) is an optional parameter that replays samples at an alternate speed than what the data was collected. 0, the default value for the parameter, specifies that every sample is played back during the GazeReplay. This value can be specified in percentage text (e.g., 50%) or in decimal double notation (e.g., .5). A setting of .5 would replay the samples at half speed and therefore would take twice as long to replay. A setting of 2 would replay the samples at double the speed and would take half as long to replay.

1) ***Drag*** the **TETGazeReplay** object from the **Tobii Pro Toolbox** and ***drop*** it after the **Feedback** object in the **TrialProc.**



2) ***Open*** the object's Property Pages.

3) ***Edit*** the **Parameters** field to be **c, Stimulus, Stimulus.OnsetTime.**



Package: TET

Routine: GazeReplay

Parameters: c, Stimulus, Stimulus.OnsetTime

Details

Overview:
Visually replays the history of eye tracker samples.

Parameters:
c (Context)

The experiment context allows the user to add variables to the experiment. The routines in the PackageFile may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every PackageFile routine.

theSlide (Slide)

The name of the Slide object that is drawn on screen before replay begins. <Slide> needs

4) ***Click*** the **OK** button to accept the changes.

Task 15: Run the Experiment

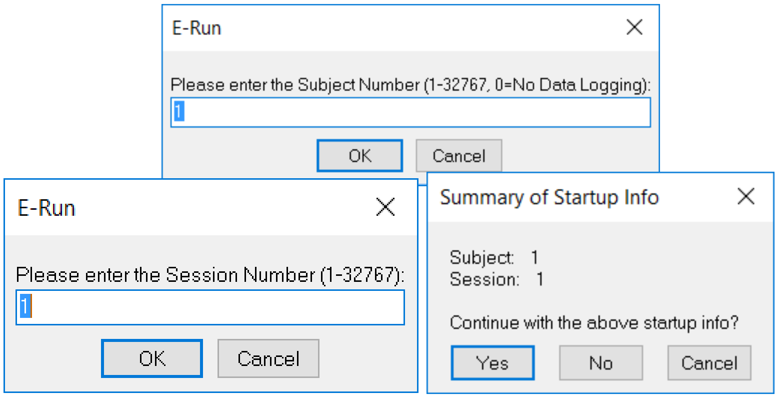
Run the experiment to verify that the eye tracker is working.

The steps necessary to create a basic EET-enabled paradigm are now completed. EET- enabled experiments can be run locally from E-Studio during development and testing.

Before testing the experimental task, it is important to understand how the task works. This experiment requires the sound on and speakers connected to the E-Prime computer because the experiment plays sounds. The task itself is simple; the name of an animal is presented simultaneously with a recording of the noise that animal makes. Next, pictures of two animals on screen are shown. One picture is on the left and one picture is on the right. If the animal on the right made the sound, the "2" key is pressed. If the animal on the left made the sound, the "1" key is pressed.

1) ***Click*** the **Run icon** on the toolbar to generate and run the experiment locally.

2) ***Press*** **Enter** to ***accept*** the **default values** for each of the initialization prompts presented.



E-Run

Please enter the Subject Number (1-32767, 0=No Data Logging):

1

E-Run

Please enter the Session Number (1-32767):

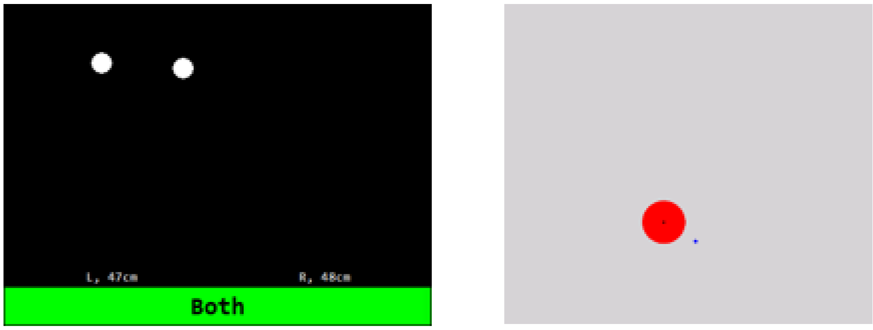
1

Summary of Startup Info

Subject: 1
Session: 1

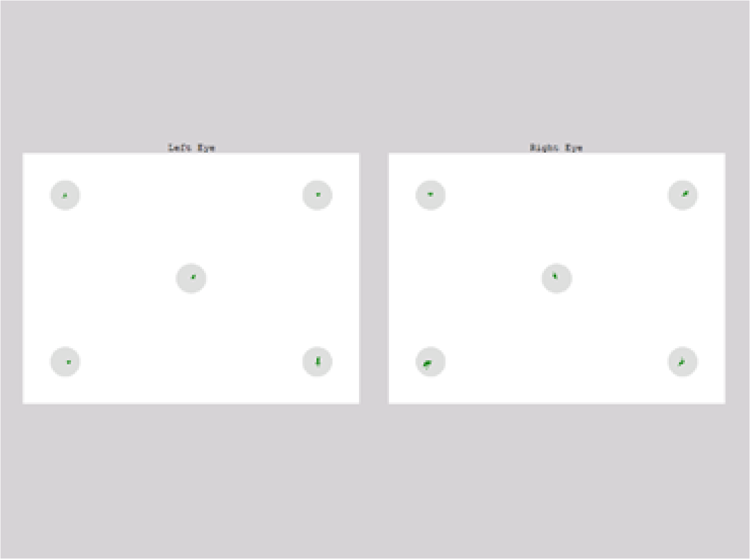
Continue with the above startup info?

2) ***Follow*** the **prompts on screen** to ***complete*** the **calibration** sequence.

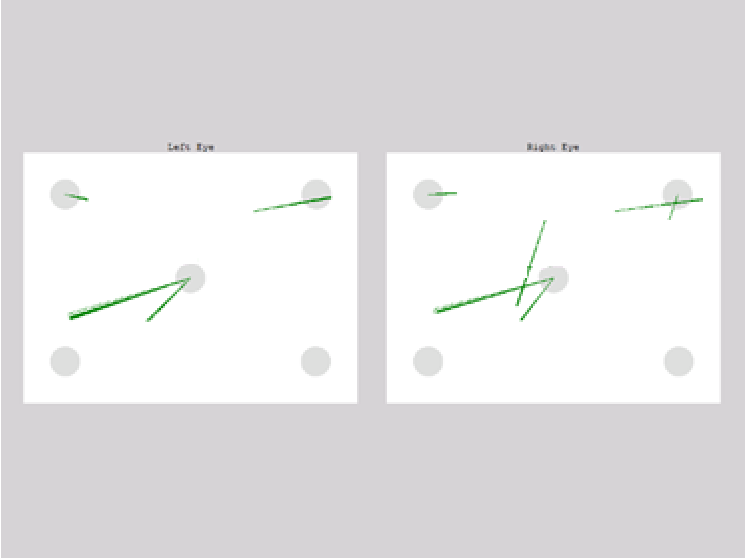


NOTE: *Once calibration is completed, a prompt occurs to verify that the calibration data is acceptable. The pictures below show the difference between a good and bad calibration. If a bad calibration is accepted, it is difficult to collect valid data during the experiment because the eye tracker cannot accurately determine where on the monitor the eyes are.*

The image below is a representation of **good calibration** data. Notice how the gaze data is within the circles (e.g., the small green dots and lines). This indicates that the eye tracker will be able to collect usable data.

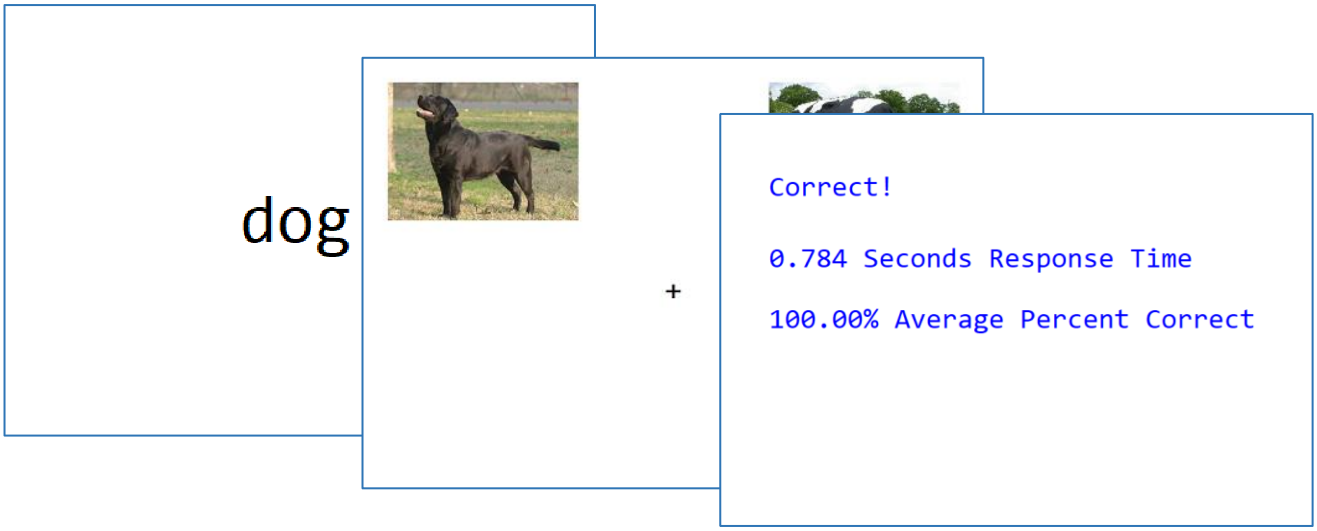


The image below is a representation of **bad calibration** data. Gaze data that leaves the circles is undesirable and is not recommended when testing. Additionally, circles that do not have green dots or small green lines inside of them are not good.



4) ***Accept*** the **calibration (if acceptable)** and ***perform*** the **experiment**.

5) ***Observe*** the **stimulus presentation sequence** to verify the experiment is functioning correctly and completes with no errors being generated.



Correct!

0.784 Seconds Response Time

100.00% Average Percent Correct

3.4. Tutorial 2: Logging Gaze Data in an Experiment

Summary:

With basic Tobii Pro support incorporated into an E-Prime experiment, the experiment now can be enabled to write the eye tracking gaze data to a file. E-Prime provides extensive capabilities for collecting, reporting, and viewing data. In most designs, E-Prime is used to collect and output one data row per trial. In eye tracking experiments, the goal is to collect and output data for every eye gaze sample (of which there may be several hundreds to thousands per trial depending on the trial duration). Because of this, EET paradigms are enabled to employ an additional data collection and output technique which results in the creation of a tab delimited text file. This file contains data rows which are a combination of eye gaze data received by E-Prime from the eye tracker and any additional behavioral data added by the user via E-Prime.

NOTE: *The typical E-Prime data file (.edat3) generated by an experiment is always available (unless you take steps to specify otherwise). For most eye tracking studies, the output file described in this section is the data most used in eye tracking data analysis (as opposed to the .edat3 file). It is up to the experimenter to verify that the correct data is available for the analysis which the experimenter expects to perform on the data.*

NOTE: *This tutorial assumes that Tutorial 1: Adding Tobii Pro Support to an E-Prime Experiment is completed and the experiment edited from that tutorial is available (i.e.,**TETFixedPositionAOI.es3**).*

NOTE: *The Sample experiment found in the folder …My Experiments\3.0\Tobii Pro\Samples\TET\ TETFixedPositionAOI.es3 represents the complete set of changes documented in Tutorials 1 and 2.*

NOTE: *If you do not want to overwrite your previous work, save the experiment under a different name before you begin this tutorial.*

Goal:

This tutorial illustrates how to write data to a Tab Delimited file. Once you completed this tutorial, gaze data can be saved from an EET-enabled experiment.

Overview of Tasks:

- Add the TET PackageCall to open the gaze data file.
- Add the TET PackageCall to close the gaze data file.
- Add the TET PackageCall to add user defined data to the gaze data file.
- Add the TET PackageCall to save the gaze data.
- Verify the overall experiment structure and run the experiment.
- Open the gaze data file.
- Review the values logged in the gaze data file and the Validity Ratings.

Estimated Tutorial Time:

10-15 minutes

Task 1: Add the TETGazeDataOpen PackageCall to Create and Open the Gaze Data File

Add a PackageCall to the SessionProc that opens the GazeData.txt file.

The first step is to add the **TETGazeDataOpen** PackageCall after the TETCalibRegular PackageCall. The purpose of this call is to open the Tab Delimited GazeData.txt file that is created once an experiment has completed. The name of the gaze data file defaults to [DataFile.BaseName]-GazeData.txt. DataFile.BaseName defaults to [ExperimentName]-[Subject#]-[Session#]. For example, the file could be named TETFixedPositionAOI-1-1-GazeData.txt (where Subject = 1, Session = 1).

The first parameter is “**c**.” It is required and passes in the current experiment Context.

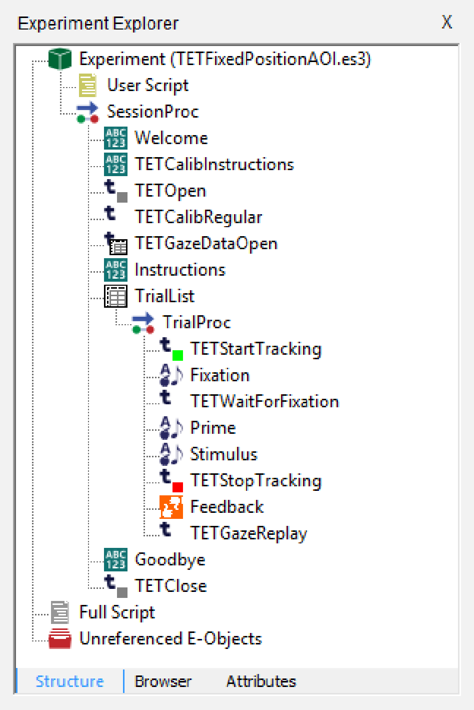
The next parameter (**vUserPropertyList**) is optional. If specified, a column for the user defined property appears in the gaze data file. Multiple columns can be specified by using a semi-colon. For example, to add two new user specified columns to the GazeData file named “Prime” and “AOI”, “Prime;AOI” would be specified. This parameter is used along with TETGazeDataSetProperty which specifies what data is saved to the column headings (see *Task 3: Add the TETGazeDataSetProperty PackageCall to add User Defined data to the Gaze Data file*).

The next parameter (**vFilename**) is an optional parameter. If specified, this parameter sets the file name of the gaze data file to the contents of this property. If this is not specified, the default file name is used (i.e., [DataFile.BaseName]-GazeData.txt).

The next parameter (**vUnicode**) is an optional parameter that enables Unicode text if specified as True. The default value in the PackageCall is blank which is equivalent to False.

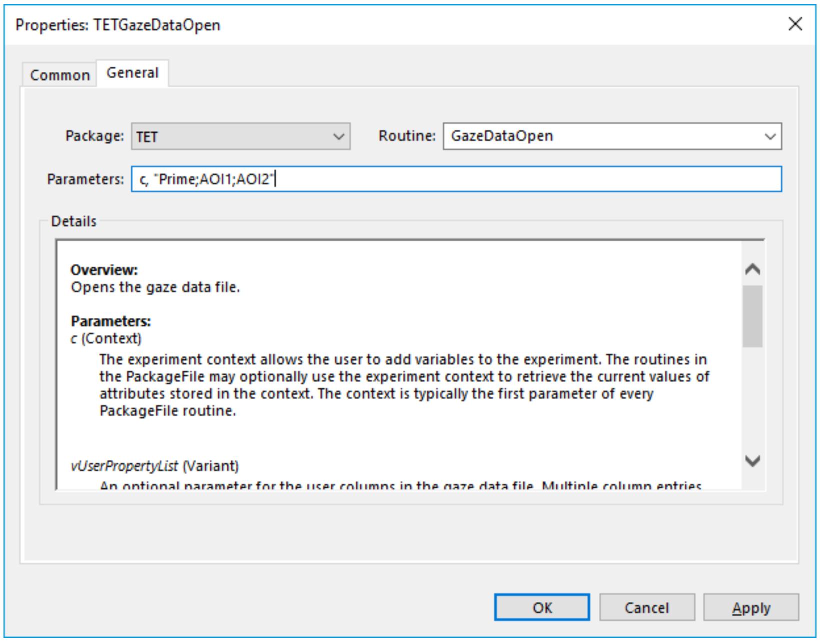
NOTE: *If it is not already opened, open the TETFixedPoisitionAOI.es3 experiment edited in the previous tutorial.*

1) ***Drag*** the **TETGazeDataOpen** object from the **Tobii Pro Toolbox** and ***drop*** it ***after*** **TETCalibRegular** in the **SessionProc**.



2) ***Open*** the object's **Property Pages**.

3) ***Edit*** the **Parameters** field to be **c, "Prime;AOI1;AOI2"** to add the user-defined columns of "**Prime**", "**AOI1**", and "**AOI2**" to the gaze data file.



Package: TET

Routine: GazeDataOpen

Parameters: c, "Prime;AOI1;AOI2"

Details

Overview:
Opens the gaze data file.

Parameters:
c (Context)

The experiment context allows the user to add variables to the experiment. The routines in the PackageFile may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every PackageFile routine.

vUserPropertyList (Variant)

An optional parameter for the user columns in the gaze data file. Multiple column entries

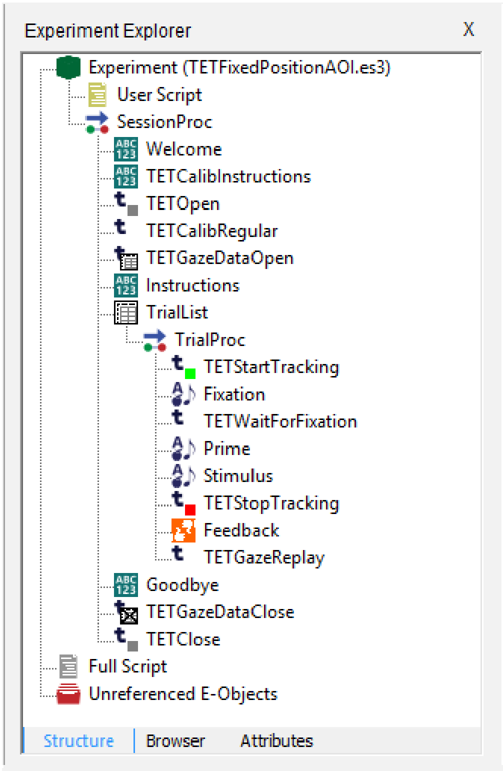
4) ***Click*** the **OK** button to accept the changes.

Task 2: Add the TETGazeDataClose PackageCall to End Data Collection

Add a PackageCall at the end of the SessionProc that closes the gaze data *file.*

TETGazeDataClose needs to be added to the end of the experiment to close the gaze data file that was previously opened. This call should be made only after eye tracking data collection with the experiment is complete.

1) ***Drag*** the **TETGazeDataClose** object from the **Tobii Pro Toolbox** and ***drop*** it ***after*** the **Goodbye** object in the **SessionProc.**



- Experiment (TETFixedPositionAOI.es3)
  - User Script
  - SessionProc
    - Welcome
    - TETCalibInstructions
    - TETOpen
    - TETCalibRegular
    - TETGazeDataOpen
    - Instructions
    - TrialList
      - TrialProc
        - TETStartTracking
        - Fixation
        - TETWaitForFixation
        - Prime
        - Stimulus
        - TETStopTracking
        - Feedback
        - TETGazeReplay
    - Goodbye
    - TETGazeDataClose
    - TETClose
  - Full Script
  - Unreferenced E-Objects

Task 3: Add the TETGazeDataSetProperty PackageCall to add User Defined data to the Gaze Data file

Add a PackageCall at the beginning of the TrialProc that adds user data to the gaze data file.

The PackageCall **TETGazeDataSetProperty** allows user data to be specified that later appear in the finished gaze data file. Users need to specify both the name and the value of the property. These values are written to their pre-defined columns in the gaze data file. In this tutorial, the columns are “Prime”, “AOI1”, and “AOI2” as discussed in *Task 1: Add the TETGazeDataOpen PackageCall to Create and Open the Gaze Data File.*

The first parameter is the current experiment **Context**.

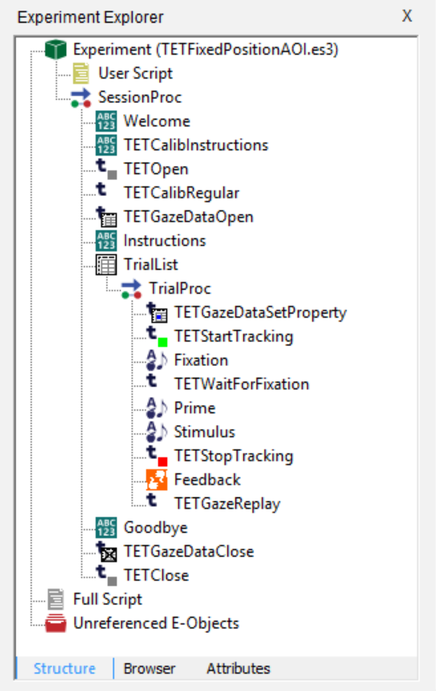
The second parameter (**property**) is required. This parameter allows for the specification of which column should be written to in the gaze data file. This property works in conjunction with the vUserPropertyList specified in TETGazeDataOpen (see *Task 1: Add the TETGazeDataOpen PackageCall to Create and Open the GazeData File*). For example, if “Prime;AOI” was specified in TETGazeDataOpen, then “property” should be specified as “Prime” to write values to the Prime column.

The third parameter (**value**) is required. This is a string parameter that is the value that is going to be seen in the gaze data file. The value appears under the corresponding column header. For example, if an attribute named “Image” exists in the experiment with the values of “dog,” “cat,” and “bird” then [Image] could be specified in this parameter. The values of “dog,” “cat,” and “bird” would appear in the gaze data file for the trials on which they are the value of the attribute (e.g., “dog” appears for trial 1 eye tracking gaze data, “cat” for trial 2 gaze data, etc.).

The rest of the parameters are all optional. These parameters (e.g., **property02, value02,** etc**.**) are necessary if additional columns and attributes are needed in the gaze data file. These parameters work identically to the related parameter mentioned above. A total of 10 sets of properties and values can be added with a single TETGazeDataSetProperty PackageCall. However, more than 10 can be added if additional TETGazeDataSetProperty PackageCalls are added to the experiment structure.

NOTE: *The PackageCalls TETGazeDataSetPropertyMatch and TETGazeDataSetPropertyNumeric exist and work similarly to TETGazeDataSetProperty. They function similarly to conditional statements (e.g., If…Then statements). For more information, please view the PackageCall descriptions in E-Studio.*

1) ***Drag*** the **TETGazeDataSetProperty** object from the **Tobii Pro Toolbox** and ***drop*** it as the **first** object in the **TrialProc.**

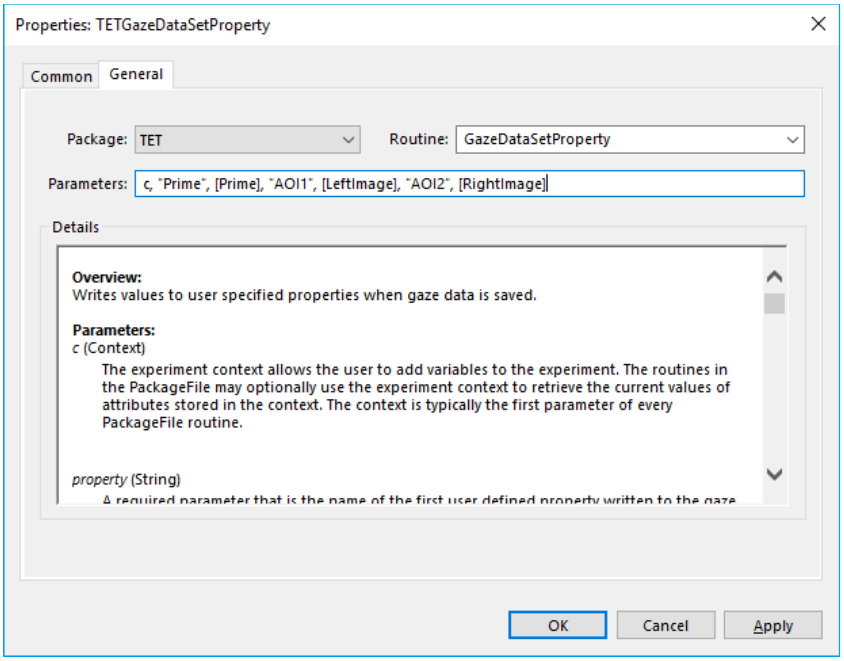


Experiment Explorer

- Experiment (TETFixedPositionAOI.es3)
  - User Script
  - SessionProc
    - Welcome
    - TETCalibInstructions
    - TETOpen
    - TETCalibRegular
    - TETGazeDataOpen
    - Instructions
    - TrialList
      - TrialProc
        - TETGazeDataSetProperty
        - TETStartTracking
        - Fixation
        - TETWaitForFixation
        - Prime
        - Stimulus
        - TETStopTracking
        - Feedback
        - TETGazeReplay
    - Goodbye
    - TETGazeDataClose
    - TETClose
- Full Script
- Unreferenced E-Objects

2) ***Open*** the object's **Property Pages**.

3) ***Edit*** the **Parameters** field to be **c, "Prime", [Prime], "AOI1", [LeftImage], "AOI2", [RightImage].**



Properties: TETGazeDataSetProperty

Package: TET

Routine: GazeDataSetProperty

Parameters: c, "Prime", [Prime], "AOI1", [LeftImage], "AOI2", [RightImage]

Details

Overview:
Writes values to user specified properties when gaze data is saved.

Parameters:
c (Context)

The experiment context allows the user to add variables to the experiment. The routines in the PackageFile may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every PackageFile routine.

property (String)

A required parameter that is the name of the first user defined property written to the gaze

4) ***Click*** the **OK** button to accept the changes.

Task 4: Add the TETGazeDataSave PackageCall to Save Eye Tracking Data

Add a PackageCall to the TrialProc that saves stimulus data to the GazeData.txt file.

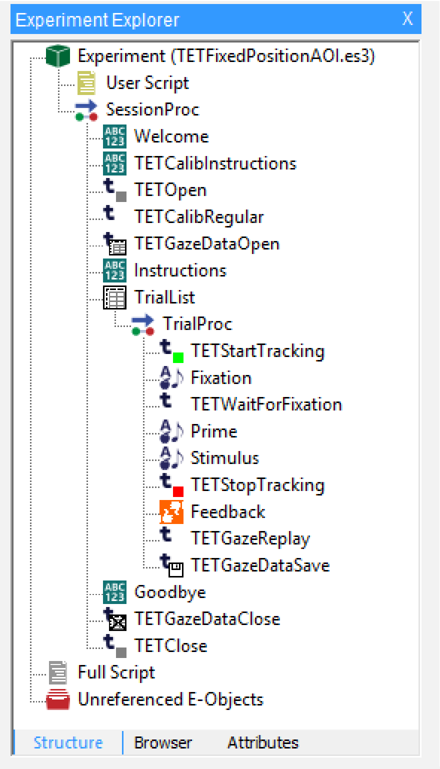
TETGazeDataSave saves the gaze and user specified data to the gaze data file. This PackageCall should be placed at the location in the experiment where gaze data is to be saved. This is typically done at the end of a trial.

The first parameter is the current experiment **Context**.

The second parameter (**vStimulusName**) is optional. If specified, a stimulus presentation object's ACC, CRESP, RESP, and RT is logged in the gaze data file.

The last parameter (**vClearHistory)** is optional with a default value of true. The purpose of the parameter is to clear the history of accumulated eye tracking samples after the gaze data is saved.

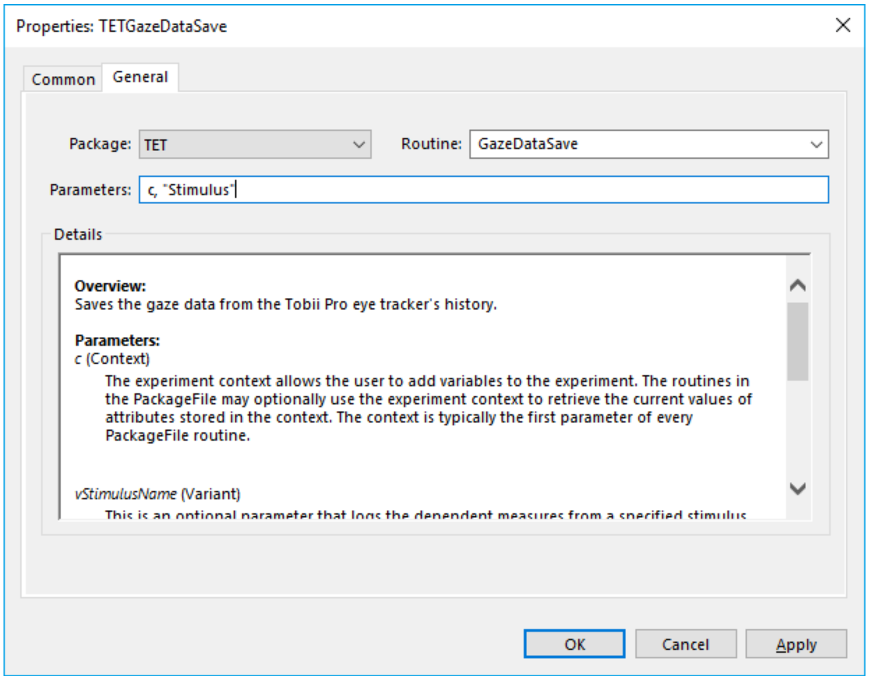
1) ***Drag*** the shortcut **TETGazeDataSave** object from the **Tobii Pro Toolbox** and ***drop*** it after **TETGazeReplay** in the **TrialProc.**



```
Experiment (TETFixedPositionAOI.es3)
    User Script
    SessionProc
        Welcome
        TETCalibInstructions
        TETOpen
        TETCalibRegular
        TETGazeDataOpen
        Instructions
        TrialList
            TrialProc
                TETStartTracking
                Fixation
                TETWaitForFixation
                Prime
                Stimulus
                TETStopTracking
                Feedback
                TETGazeReplay
                TETGazeDataSave
        Goodbye
        TETGazeDataClose
        TETClose
    Full Script
    Unreferenced E-Objects
```

2) ***Open*** the object's **Property Pages**.

3) ***Edit*** the **Parameters** field to be **c,** "**Stimulus**".



Package: TET

Routine: GazeDataSave

Parameters: c, "Stimulus"

Overview:
Saves the gaze data from the Tobii Pro eye tracker's history.

Parameters:
c (Context)

The experiment context allows the user to add variables to the experiment. The routines in the PackageFile may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every PackageFile routine.

vStimulusName (Variant)

This is an optional parameter that logs the dependent measures from a specified stimulus

4) ***Click*** the **OK** button to accept the changes.

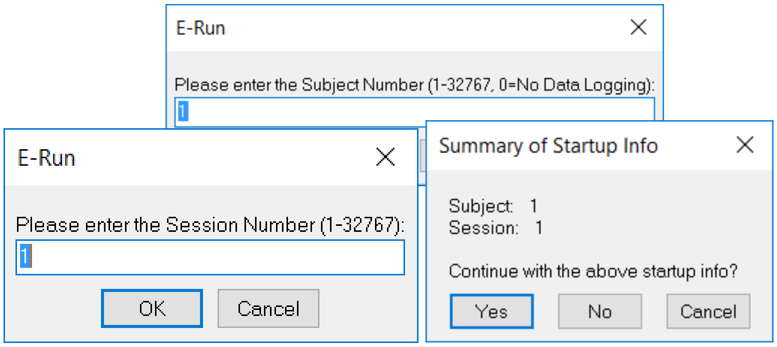
Task 5: Run the Experiment

Run the experiment to verify that the eye tracker is working and to ensure the GazeData.txt file is written.

With the additional gaze data PackageCalls added in the experiment, a gaze data file can be generated for a test session. The gaze data file created at the end of this experiment run contains the typical Tobii data (discussed in *Task 7: Introduction to Gaze Data File*), and the Stimulus object and attribute data.

1) ***Click*** the **Run icon** on the toolbar to generate and run the experiment locally.

2) ***Press*** **Enter** to ***accept*** the **default values** for each of the initialization prompts presented.



E-Run

Please enter the Subject Number (1-32767, 0=No Data Logging):

1

E-Run

Please enter the Session Number (1-32767):

1

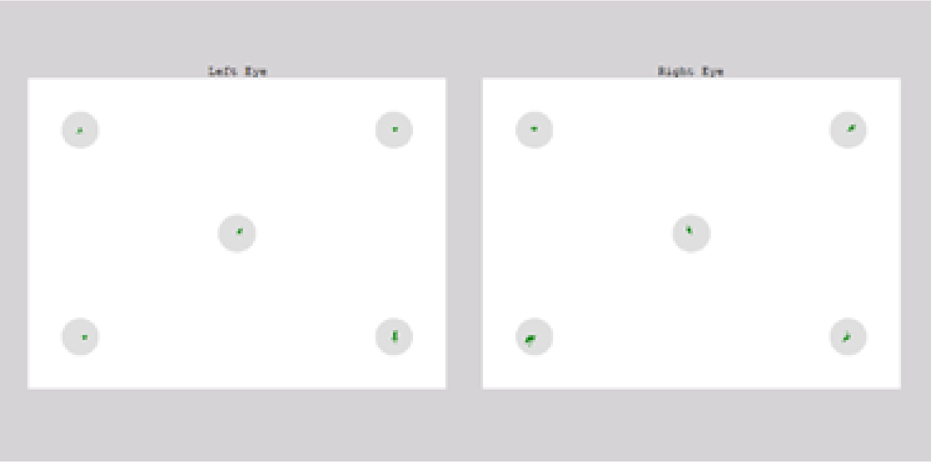
Summary of Startup Info

Subject: 1
Session: 1

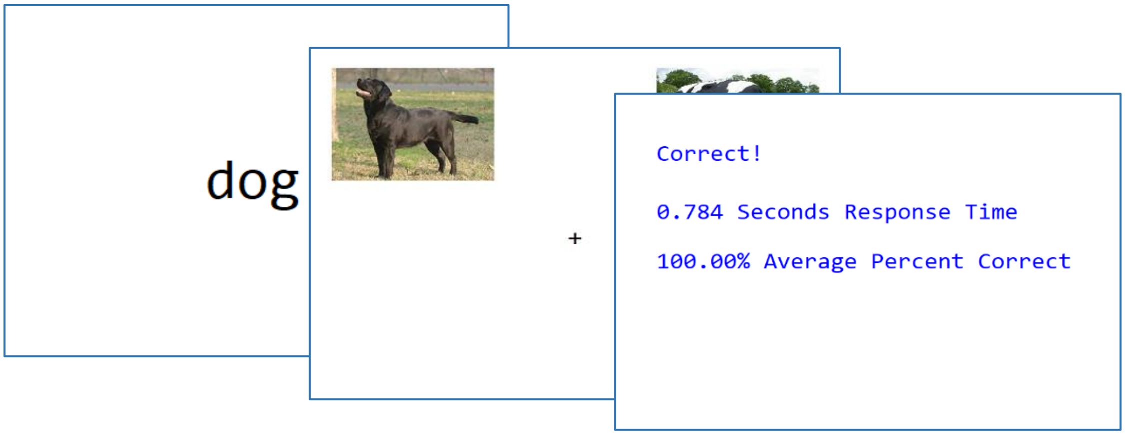
Continue with the above startup info?

3) ***Follow*** the **prompts on screen** to ***complete*** the **calibration** sequence.

4) ***Accept*** the **calibration (if acceptable)** and ***perform*** the **experiment**.



5) ***Observe*** the **stimulus presentation sequence** to ***verify*** the **experiment** is ***functioning*** correctly and ***completes*** with **no errors** being generated.



Correct!

0.784 Seconds Response Time

100.00% Average Percent Correct

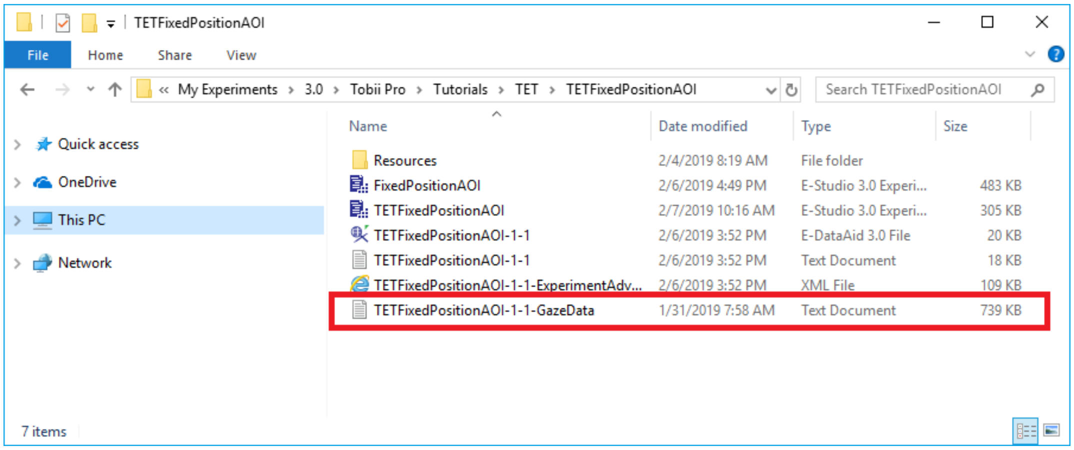
Task 6: Open the Gaze Data File

Verify the gaze data file exists and can be opened.

The gaze data file contains information about the stimulus presentation (e.g., what was on screen and when it was on screen, the eye tracking data, where the eyes were looking). This file facilitates data analysis. After the experiment is created, it is important that you check this file thoroughly to make sure all the components you need for your data analysis are included in the file before you begin running study participants.

NOTE: *The file name varies depending on what subject number and session number you chose to run the experiment under.*

1) ***Navigate*** to **…My Experiments\3.0\Tobii Pro\Tutorials\TET\TETFixedPositionAOI\ TETFixedPositionAOI-1-1-GazeData**.**txt**.

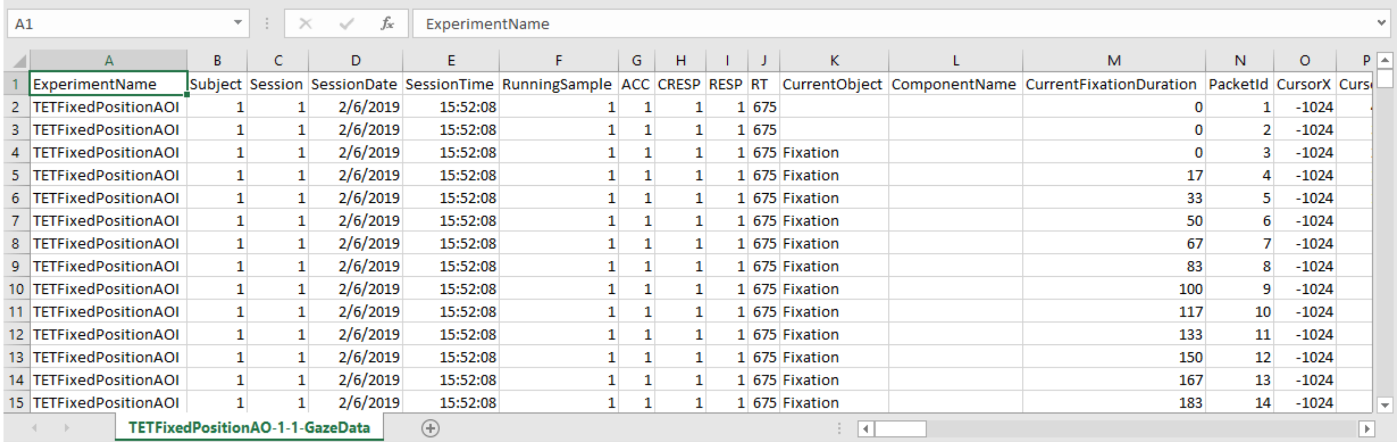

TETFixedPositionAOI
File
Home
Share
View
« My Experiments › 3.0 › Tobii Pro › Tutorials › TET › TETFixedPositionAOI
Search TETFixedPositionAOI
Quick access
OneDrive
This PC
Network
7 items

Name	Date modified	Type	Size
Resources	2/4/2019 8:19 AM	File folder	
FixedPositionAOI	2/6/2019 4:49 PM	E-Studio 3.0 Experi...	483 KB
TETFixedPositionAOI	2/7/2019 10:16 AM	E-Studio 3.0 Experi...	305 KB
TETFixedPositionAOI-1-1	2/6/2019 3:52 PM	E-DataAid 3.0 File	20 KB
TETFixedPositionAOI-1-1	2/6/2019 3:52 PM	Text Document	18 KB
TETFixedPositionAOI-1-1-ExperimentAdv...	2/6/2019 3:52 PM	XML File	109 KB
TETFixedPositionAOI-1-1-GazeData	1/31/2019 7:58 AM	Text Document	739 KB

NOTE: *Additionally, the gaze data file can be opened by clicking on the hyperlink listed in the Output window in E-Studio.*

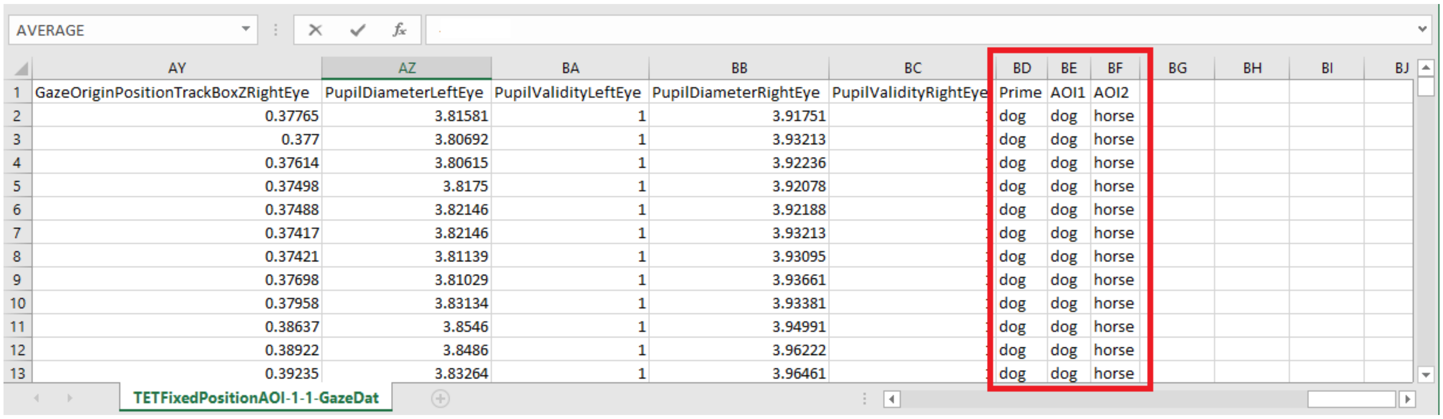
2) ***Right click*** the **TETFixedPositionAOI-1-1-GazeData.txt file**. ***Select*** **Open With > Microsoft Office Excel**. If prompted, accept all defaults. ***Click*** **Finish**.

3) ***Verify*** that the **file** opens and looks like the image below.



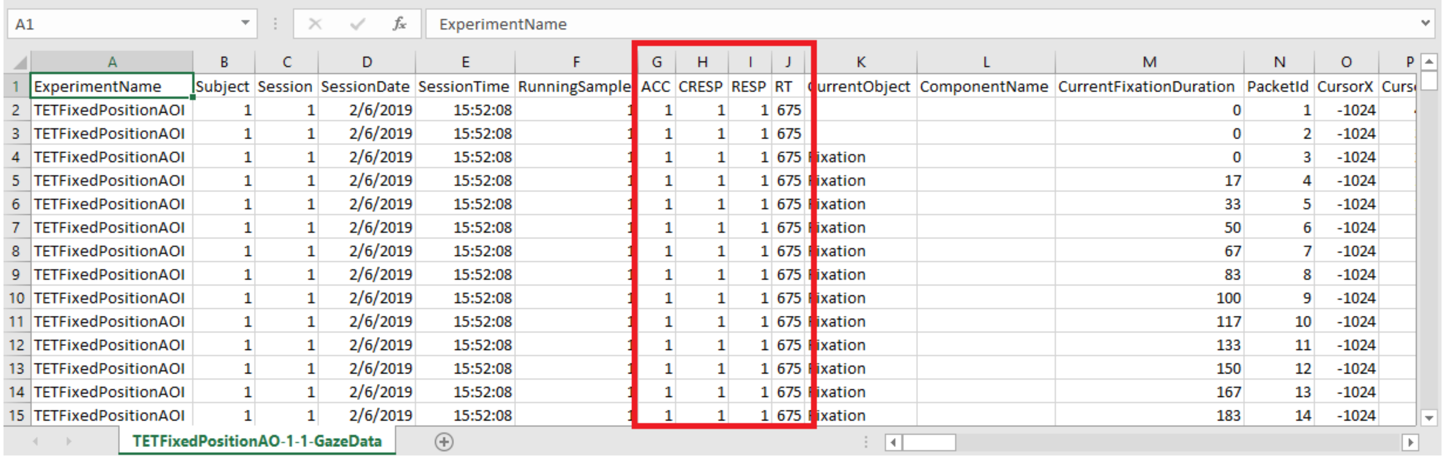
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	ExperimentName	Subject	Session	SessionDate	SessionTime	RunningSample	ACC	CRESP	RESP	RT	CurrentObject	ComponentName	CurrentFixationDuration	PacketId	CursorX	Curs
2	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675			0	1	-1024	
3	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675			0	2	-1024	
4	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		0	3	-1024	
5	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		17	4	-1024	
6	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		33	5	-1024	
7	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		50	6	-1024	
8	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		67	7	-1024	
9	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		83	8	-1024	
10	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		100	9	-1024	
11	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		117	10	-1024	
12	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		133	11	-1024	
13	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		150	12	-1024	
14	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		167	13	-1024	
15	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		183	14	-1024	

4) ***Verify*** the **user defined values and columns** logged in *Task 3* are present in the gaze data file.



	AY	AZ	BA	BB	BC	BD	BE	BF	BG	BH	BI	BJ
1	GazeOriginPositionTrackBoxZRightEye	PupilDiameterLeftEye	PupilValidityLeftEye	PupilDiameterRightEye	PupilValidityRightEye	Prime	AOI1	AOI2				
2	0.37765	3.81581	1	3.91751	1	dog	dog	horse				
3	0.377	3.80692	1	3.93213	1	dog	dog	horse				
4	0.37614	3.80615	1	3.92236	1	dog	dog	horse				
5	0.37498	3.8175	1	3.92078	1	dog	dog	horse				
6	0.37488	3.82146	1	3.92188	1	dog	dog	horse				
7	0.37417	3.82146	1	3.93213	1	dog	dog	horse				
8	0.37421	3.81139	1	3.93095	1	dog	dog	horse				
9	0.37698	3.81029	1	3.93661	1	dog	dog	horse				
10	0.37958	3.83134	1	3.93381	1	dog	dog	horse				
11	0.38637	3.8546	1	3.94991	1	dog	dog	horse				
12	0.38922	3.8486	1	3.96222	1	dog	dog	horse				
13	0.39235	3.83264	1	3.96461	1	dog	dog	horse				

5) ***Verify*** the **Stimulus object data** logged in *Task 4* are present in the gaze data file.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	ExperimentName	Subject	Session	SessionDate	SessionTime	RunningSample	ACC	CRESP	RESP	RT	CurrentObject	ComponentName	CurrentFixationDuration	PacketId	CursorX	Curs
2	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675			0	1	-1024	
3	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675			0	2	-1024	
4	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		0	3	-1024	
5	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		17	4	-1024	
6	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		33	5	-1024	
7	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		50	6	-1024	
8	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		67	7	-1024	
9	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		83	8	-1024	
10	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		100	9	-1024	
11	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		117	10	-1024	
12	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		133	11	-1024	
13	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		150	12	-1024	
14	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		167	13	-1024	
15	TETFixedPositionAOI	1	1	2/6/2019	15:52:08	1	1	1	1	675	Fixation		183	14	-1024	

Task 7: Introduction to Gaze Data File

Understand the general output from the Gaze Data.txt file.

The gaze data file contains information about eye movements from the Tobii Pro Eye Tracker and information about the stimulus presentation from E-Prime. The data consists of general information regarding the eye gaze data that is continuously collected as well as user-defined data. This data is compiled in one file to make analysis easier. The table below summarizes the general information included in the gaze data file. Take some time to familiarize yourself with the basic output and the information that it provides. This way you can determine what you need to add to the data file in the way of user defined columns.

Gaze Data File	
Column	***Definition***
ExperimentName	The name of the experiment that generates the gaze data file.
Subject	The Subject number (from E-Prime Startup Info).
Session	The Session number (from E-Prime Startup Info).
SessionDate	The date the test session is run.
SessionTime	The time the test session is run.
LineNumber	The line number in the gaze data file.
RunningSample	The number of sample that is currently running
ACC	The accuracy of the response a specified InputMask.
CRESP	The correct response for a specified InputMask.
RESP	The response made with a specified InputMask.
RT	The reaction time associated with the last response processed by an InputMask.
CurrentObject	The name of the current object in the experiment that is being looked at by the participant (e.g., the name of a Slide).
ComponentName	The name of the component in the experiment that is being looked at by the participant (e.g., the name of a SlideImage).
CurrentFixationDuration	The amount of time a fixation occurs during the CurrentObject and ComponentName.
CursorX	The X axis pixel location that corresponds to where the gaze is located on the screen and accounts for screen resolution (width).
CursorY	The Y axis pixel location that corresponds to where the gaze is located on the screen and accounts for screen resolution (height).
RTTime	The timestamp value based on E-Prime's clock. Returns the time stamp of when it currently is in the experiment.
RTTimeMicro	The microsecond timestamp value based on E-Prime's clock. Returns the time stamp of when it currently is in the experiment.
RTTimeReceived	The E-Prime time when the application received the eye tracker packet.
RTTimeReceivedMicro	The E-Prime time in microseconds when the application received the eye tracker packet.
SystemTimestamp	The complete system timestamp provided within the eye tracker sample (reported in microseconds).
HardwareTimestamp	The complete device timestamp provided within the eye tracker sample (reported in microseconds).
HostTimestamp	The eye tracker system time when the application receives the gaze data (reported in microseconds).

Gaze Data File continued	
Column	***Definition***
SEQD	The sequence of the response provided by the device
SEQH	The sequence of the response provided by the InputHistoryManager.
GazePointValidityLeftEye	The validity of the left eye gaze point data.
GazePointPositionDisplayXLeftEye	The horizontal position (X) of the left eye gaze point on the active display (0, 0 is the upper left corner and 1, 1 is the lower right corner).
GazePointPositionDisplayYLeftEye	The vertical position (Y) of the left eye gaze point on the display (0, 0 is the upper left corner and 1, 1 is the lower right corner).
GazePointValidityRightEye	The validity of the right eye gaze point data.
GazePointPositionDisplayXRightEye	The horizontal position (X) of the right eye gaze point on the display (0, 0 is the upper left corner and 1, 1 is the lower right corner).
GazePointPositionDisplayYRightEye	The vertical position (Y) of the right eye gaze point on the display (0, 0 is the upper left corner and 1, 1 is the lower right corner).
GazePointPositionUserXLeftEye	The horizontal gaze point position (X) in the user coordinate system for the left eye (the x-axis points horizontally towards the user’s right).
GazePointPositionUserYLeftEye	The vertical gaze point position (Y) in the user coordinate system for the left eye (the y-axis points vertically towards the ceiling).
GazePointPositionUserZLeftEye	The gaze point position from the user to the eye tracker (Z) in the user coordinate system for the left eye (the z-axis points towards the user).
GazePointPositionUserXRightEye	The horizontal gaze point position (X) in the user coordinate system for the right eye (the x-axis points horizontally towards the user’s right).
GazePointPositionUserYRightEye	The vertical gaze point position (Y) in the user coordinate system for the right eye (the y-axis points vertically towards the ceiling).
GazePointPositionUserZRightEye	The gaze point position from the user to the eye tracker (Z) in the user coordinate system for the right eye (the z-axis points towards the user).
GazeOriginValidityLeftEye	The validity of the left eye gaze origin data.
GazeOriginPositionUserXLeftEye	The horizontal gaze origin position (X) in the user coordinate system for the left eye (the x-axis points horizontally towards the user’s right).
GazeOriginPositionUserYLeftEye	The vertical gaze origin position (Y) in the user coordinate system for the left eye (the y-axis points vertically towards the ceiling).
GazeOriginPositionUserZLeftEye	The gaze origin position from the user to the eye tracker (Z) in the user coordinate system for the left eye (the z-axis points towards the user).
GazeOriginValidityRightEye	The validity of the right eye gaze origin data.
GazeOriginPositionUserXRightEye	The horizontal gaze origin position (X) in the user coordinate system for the right eye (the x-axis points horizontally towards the user’s right).
GazeOriginPositionUserYRightEye	The vertical gaze origin position (Y) in the user coordinate system for the right eye (the y-axis points vertically towards the ceiling).
GazeOriginPositionUserZRightEye	The gaze origin position from the user to the eye tracker (Z) in the user coordinate system for the right eye (the z-axis points towards the user).
GazeOriginPositionTrackBoxXLeftEye	The horizontal normalized gaze origin position (X) in the track box coordinate system for the left eye (the x-axis points horizontally towards the user’s right).
GazeOriginPositionTrackBoxYLeftEye	The vertical normalized gaze origin position (Y) in the track box coordinate system for the left eye (the y-axis points vertically towards the ground).
GazeOriginPositionTrackBoxZLeftEye	The normalized gaze origin position from the user to the eye tracker (Z) in the track box coordinate system for the left eye (the z-axis points towards the user).

Gaze Data File continued	
Column	***Definition***
GazeOriginPositionTrackBoxXRightEye	The horizontal normalized gaze origin position (X) in the track box coordinate system for the right eye (the x-axis points horizontally towards the user's right).
GazeOriginPositionTrackBoxYRightEye	The vertical normalized gaze origin position (Y) in the track box coordinate system for the right eye (the y-axis points vertically towards ground).
GazeOriginPositionTrackBoxZRightEye	The normalized gaze origin position from the user to the eye tracker (Z) in the track box coordinate system for the right eye (the z-axis points towards the user).
PupilDiameterLeftEye	The diameter of the left pupil in millimeters.
PupilValidityLeftEye	The validity of the left pupil gaze origin data.
PupilDiameterRightEye	The diameter of the right pupil in millimeters.
PupilValidityRightEye	The validity of the right pupil gaze origin data.
UserVariables 1, 2, 3, etc.	User variables specified in TETGazeDataOpen or TETGazeDataDefineProperty appear here.

For further information on the different coordinate systems used in gathering data for Gaze Data files, please refer to: Coordinate System in Tobii Eye Tracking.

Task 8: Validity Ratings

Understand the validity ratings for the GazeData.txt output.

The rows corresponding to the columns listed on the previous page contain the values of the output data. In some cases, the values need no explanation because they are timestamps or millimeters. In other cases, the data values are single digit numbers. When this is the case it is necessary to know the meaning assigned to the numeric value. The table below defines the meaning of the number.

Validity Ratings	
0	0 indicates that valid data was not recorded on this sample and eye. The corresponding gaze data has values of -1 that indicate a valid observation was not made.
1	1 indicates that valid data was observed for this sample and eye. The corresponding gaze data has appropriate values (e.g., 0.222436338) and -1 is not reported.

3.5. Tutorial 3: Multiple Monitors: Creating a Participant Station

Summary:

With the TET PackageCalls, various parts of an experiment can be displayed on more than one monitor. This allows for a creation of a Tobii Pro Display and an Experimenter Display. The **Tobii Pro Display** presents what the participant views such as the calibration and experimental stimuli. The **Experimenter Display** prevents the participant from seeing the "housekeeping" aspects of the experiment (e.g., calibration results, participant information entry screens, etc.). This is ideal for infant studies or other studies where these aspects of the experiment can distract the participant.

Display settings and video cards may differ on each machine. To configure the displays correctly, the cable positions from the video card and/or display indexes for the Tobii Pro and Experimenter Display may need switched. The display index can be determined either by viewing the machine's Display Settings dialog or by running dxdiag.exe (refer to *Article 19550: Running DXDiag)*.

It is important that the operating system sets the Tobii Pro monitor as the main display (different from display index). This makes the upper left corner of the Tobii Pro monitor the 0,0 coordinate which is necessary to correctly log the x,y eye coordinates with the gaze data file.

Once the Tobii Pro Display is set as the main display, it is necessary to identify which display index it is set as. That index then needs to be used as the Tobii Pro monitor in the experiment. Be advised that the type of cables used to connect to the video card can also affect which display the computer detects as display index 1 and 2. We advise using a dual headed video card and connecting the displays via which ever connectors are native to that video card (e.g., HDMI to HDMI).

During this tutorial, a Display Device is added to the previously altered TETFixedPositionAOI.es3 experiment. A script is created that assigns different aspects of existing E-Prime Extensions for Tobii Pro PackageCalls to different displays. When completed, the Tobii Pro Display asks to calibrate the Tobii Pro Eye Tracker and the results of the calibration are shown on the Experiment Display for analysis and approval. Upon approval, instructions are shown on the Tobii Pro Display, followed by the fixations and stimuli. Once all the trials have been completed, a Goodbye screen is shown on the Tobii Pro Display and the experiment terminates.

NOTE: *This tutorial assumes that Tutorial 1: Adding Tobii Pro Support to an E-Prime Experiment and Tutorial 2: Logging Gaze Data in an Experiment are completed and the experiment edited from those tutorials is available (i.e., **TETFixedPositionAOI.es3**).*

NOTE: *The sample experiment found in the folder …My Experiments\3.0\Tobii Pro\Samples\TET TETFixedPositionAOIMultiMonitor.es3 represents the complete set of changes documented in Tutorials 1 through 3.*

NOTE: *If you do not want to overwrite your previous work, save the experiment under a different name before you begin this tutorial.*

NOTE: *If using only 1 monitor, it is not necessary to do this tutorial.*

Goal:

This tutorial illustrates how to add multiple display capabilities to the TETFixedPositionAOI.es3. A paradigm that displays different aspects of the experiment on different displays is created.

Overview of Tasks:

- Rename the existing Display Device in the Experiment Object.
- Add a second Display Device to the Experiment Object.
- Edit the properties of the second Display Device.
- Move the second Display Device to a higher position in the Experiment Object Properties.
- Edit the TobiiEyeTracker Display Setting.
- Add an InLine Object to display desired items on each monitor.
- Verify the overall experiment structure and run the experiment.

Recommended readings:

We recommend reading the information on multiple video displays:

Article 22724: Multiple Video Displays

Estimated Tutorial Time:

10-15 minutes

Task 1: Rename the existing Display Device in the Experiment Object

Open the Property Pages for the Experiment Object and rename the Display device to “ExperimenterDisplay”

The name of the default Display Device needs to be changed from “Display” to “ExperimenterDisplay” to distinguish which Display Device is used to display different aspects of the experiment. The ExperimenterDisplay will display the “housekeeping” aspects of the experiment (e.g., calibration results).

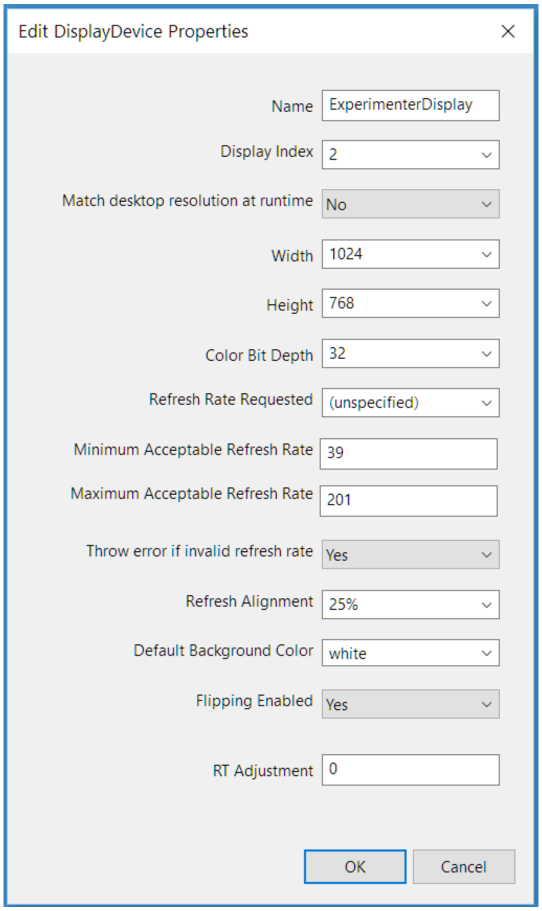
NOTE: *If it is not already opened, open the TETFixedPoisitionAOI.es3 experiment edited in the previous tutorials.*

1) ***Double click*** the **Experiment Object** at the top of the tree in the **Experiment Explorer** window.

2) ***Click*** on the **Devices** tab of the **Experiment Object Properties** dialog.

3) ***Click*** on the **Display** Device and then ***click*** the **Edit...** button.

4) ***Rename*** the **Display** object to “**ExperimenterDisplay**” and ***set*** the **Display Index** to **2**.



Name	ExperimenterDisplay
Display Index	2
Match desktop resolution at runtime	No
Width	1024
Height	768
Color Bit Depth	32
Refresh Rate Requested	(unspecified)
Minimum Acceptable Refresh Rate	39
Maximum Acceptable Refresh Rate	201
Throw error if invalid refresh rate	Yes
Refresh Alignment	25%
Default Background Color	white
Flipping Enabled	Yes
RT Adjustment	0

5) ***Click*** the **OK** button to accept the changes.

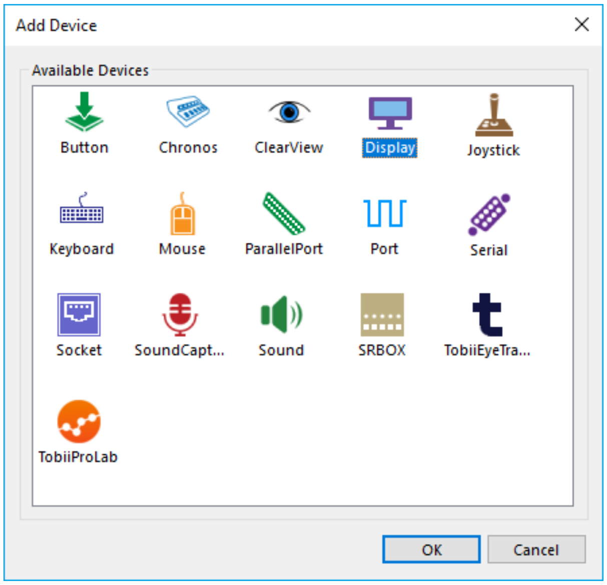
Task 2: Add a Second Display Device to the Experiment Object

Add a second Display Device to the experiment.

After editing the name of the original Display Device in the experiment, it is necessary to add a second Display Device. The second Display Device is used to present what the participant views (e.g., calibration, experimental stimuli).

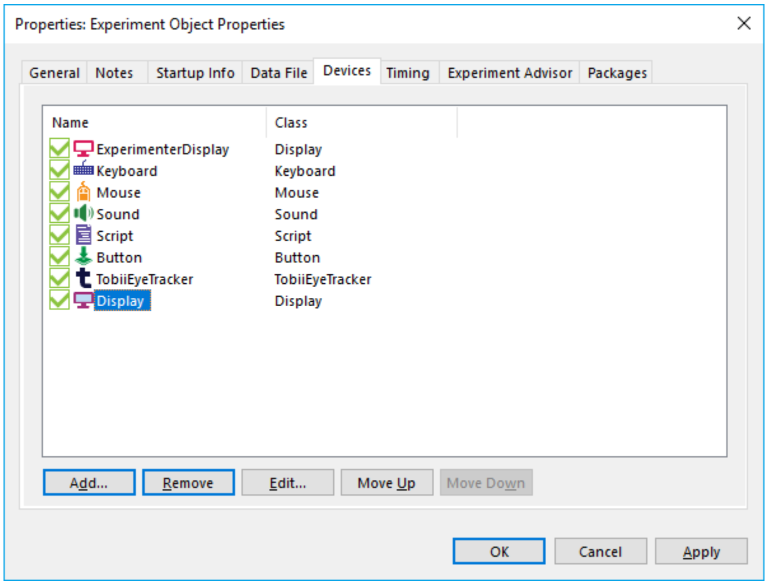
1) ***Click*** the **Add**… button.

2) ***Select*** the **Display Device** in the **Add Device dialog**.



3) ***Click*** the **OK** button to accept the changes.

4) ***Verify*** the **Display Device** is listed last under the **Name** column and is **checked**.



Name	Class
☑ ExperimenterDisplay	Display
☑ Keyboard	Keyboard
☑ Mouse	Mouse
☑ Sound	Sound
☑ Script	Script
☑ Button	Button
☑ TobiiEyeTracker	TobiiEyeTracker
☑ Display	Display

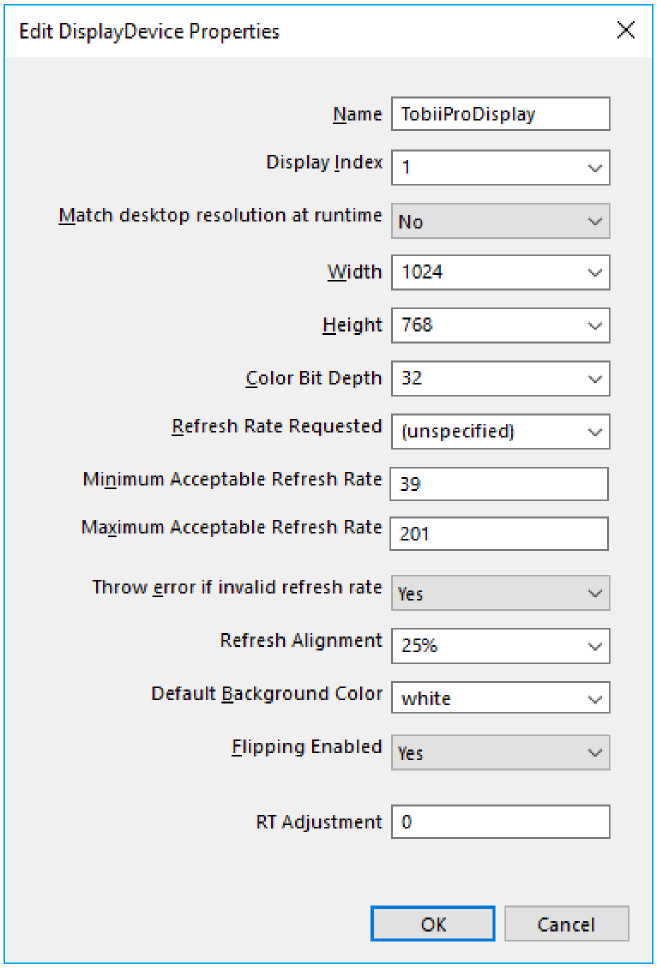
Task 3: Edit the Properties of the second Display Device

Edit the newly added second Display Device.

The properties of the new Display Device need to be changed to distinguish it from the ExperimenterDisplay. The new Display Device, the TobiiProDisplay, will display the experiment to the participant.

1) ***Click*** on the **newly added Display** Device and then ***click*** the **Edit...** button.

2) ***Rename*** the **Display** object to "**TobiiProDisplay**" and ***set*** the **Display Index** to **1**.



Name	TobiiProDisplay
Display Index	1
Match desktop resolution at runtime	No
Width	1024
Height	768
Color Bit Depth	32
Refresh Rate Requested	(unspecified)
Minimum Acceptable Refresh Rate	39
Maximum Acceptable Refresh Rate	201
Throw error if invalid refresh rate	Yes
Refresh Alignment	25%
Default Background Color	white
Flipping Enabled	Yes
RT Adjustment	0

3) ***Click*** the **OK** button to accept the changes.

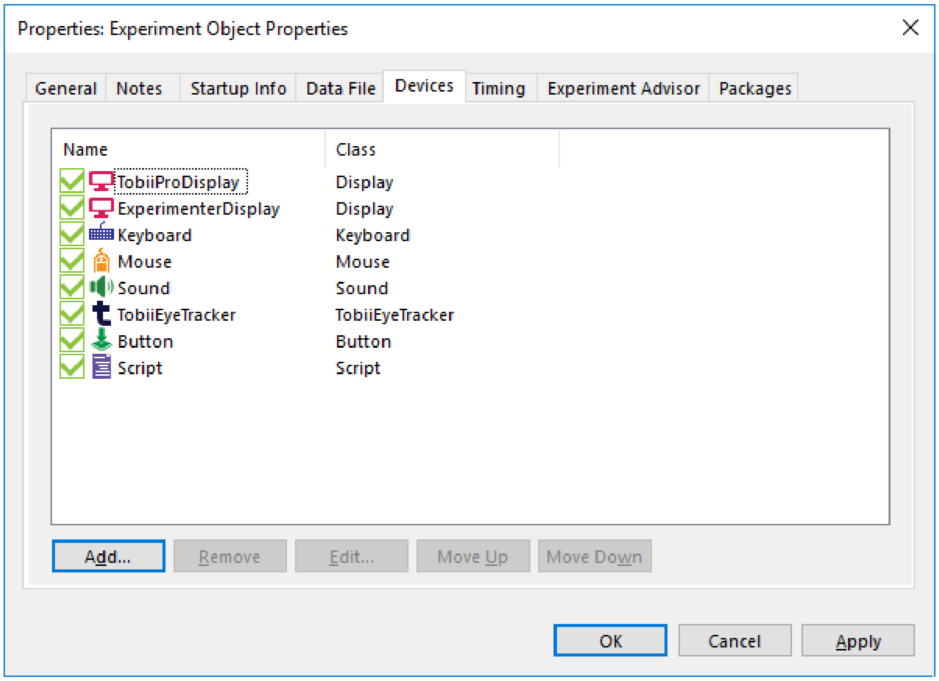
Task 4: Move the second Display Device to a Higher Position in the Experiment Object Properties

Relocate the two Display Devices.

Before exiting the Devices tab of the Experiment Object Properties, move the TobiiProDisplay Display device directly above the ExperimenterDisplay Display Device. The TobiiProDisplay device becomes the default device. The default Display device always shows the dialogs for the Subject and Session numbers. The Display Index differs for each machine and video card.

1) ***Click*** on the **TobiiProDisplay** Display Device.

2) ***Click*** on the **Move Up** button in the **Devices** tab of the **Experiment Object Properties** window until the **TobiiProDisplay** Display Device is located at the top of the list.



Name	Class
☑ TobiiProDisplay	Display
☑ ExperimenterDisplay	Display
☑ Keyboard	Keyboard
☑ Mouse	Mouse
☑ Sound	Sound
☑ TobiiEyeTracker	TobiiEyeTracker
☑ Button	Button
☑ Script	Script

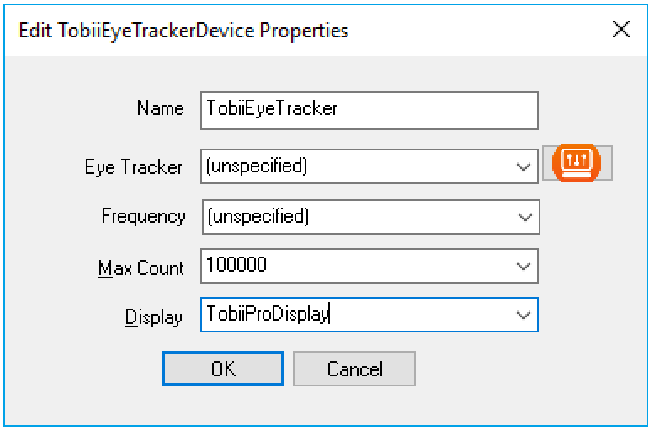
Task 5: Edit the TobiiEyeTracker Device Display Property

Edit the Display property in the TobiiEyeTracker Device for the newly added Display.

The TobiiEyeTracker Device contains a property called Display. This property needs to be set to the TobiiProDisplay which is the Display that performs the eye tracking.

1) ***Click*** on the **Display** Device and then ***click*** the **Edit...** button.

2) ***Click*** on the ***Display dropdown*** and ***specify*** **TobiiProDisplay** as the Display.



Edit TobiiEyeTrackerDevice Properties

Name: TobiiEyeTracker

Eye Tracker: (unspecified)

Frequency: (unspecified)

Max Count: 100000

Display: TobiiProDisplay

NOTE: *Verify the Name, Eye Tacker, Frequency, and Max Count properties are appropriate before continuing (e.g., specify an appropriate eye tracker).*

3) ***Click*** the **OK** button to accept the changes. ***Click*** **Apply** to commit the changes and ***click*** **OK** to dismiss the **Experiment Object Properties** window.

Task 6: Add an InLine Object to the SessionProc to display the desired items on each Monitor

Add an InLine to the SessionProc and add script to assign what is displayed on each monitor.

The **InLine** object allows to be written directly into the experiment. With it, specific functions can be placed into the experiment that the E-Studio graphical user interface may not include.

The lines of script that are written into the InLine Object allow the user to determine what part of the experiment are seen by the participant (TobiiProDisplay) or the experimenter (ExperimenterDisplay). This can be done with the TETSetMonitorDefault, TETSetMonitorCalibration, TETSetMonitorCalibrationResult, TETSetMonitorGazeReplay, and TETSetMonitorTrackStatus PackageCalls.

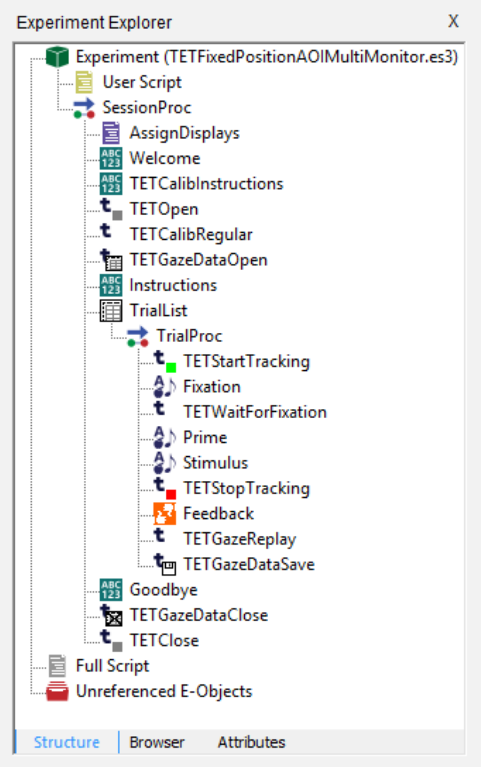
NOTE: *We typically do not recommend using PackageCalls via InLine if the PackageCall is available, however, the example below is an instance where using a PackageCall via InLine is recommended.*

If you are making direct PackageCalls via E-Basic script, please refer to:

Article 26208: Using PackageCalls via InLine Script

1) ***Drag*** the shortcut **InLine Object** from the **Toolbox** and ***drop*** it before the **Welcome** object in the **SessionProc.**

2) ***Rename*** (F2) the object **"AssignDisplays"** and ***press*** **Enter.**



```
Experiment Explorer                                   X
Experiment (TETFixedPositionAOIMultiMonitor.es3)
    User Script
    SessionProc
        AssignDisplays
        Welcome
        TETCalibInstructions
        TETOpen
        TETCalibRegular
        TETGazeDataOpen
        Instructions
        TrialList
            TrialProc
                TETStartTracking
                Fixation
                TETWaitForFixation
                Prime
                Stimulus
                TETStopTracking
                Feedback
                TETGazeReplay
                TETGazeDataSave
        Goodbye
        TETGazeDataClose
        TETClose
Full Script
Unreferenced E-Objects
Structure | Browser   Attributes
```

3) ***Open*** the **AssignDisplays InLine Object**.

4) ***Copy*** the **following text** (Ctrl+C) and ***Paste*** it (Ctrl+V) into the **AssignDisplays** InLine:

```
'This experiment is set up with a participant practice station. It uses two displays, one for the stimulus presentation
'or the participant station, and one for the experimenter. The TobiiProDisplay must be the first Display device
'in the Devices list.  This display displays the Instructions, Fixation, Stimulus, Feedback and Goodbye objects.

         TET_SetMonitorDefault "ExperimenterDisplay", True
         TET_SetMonitorTrackStatus "TobiiProDisplay", True
         TET_SetMonitorCalibration "TobiiProDisplay", True
         TET_SetMonitorCalibrationResult "ExperimenterDisplay", True
         TET_SetMonitorGazeReplay "TobiiProDisplay", True
```

NOTE: *The above text can also be copied from the TETFixedPositionAOIMultiMonitor.es3 file located in the Samples folder (…My Experiments\3.0\Tobii Pro\Samples\TET\TETFixedPositionAOIMultiMonitor).*

Task 7: Run the Experiment

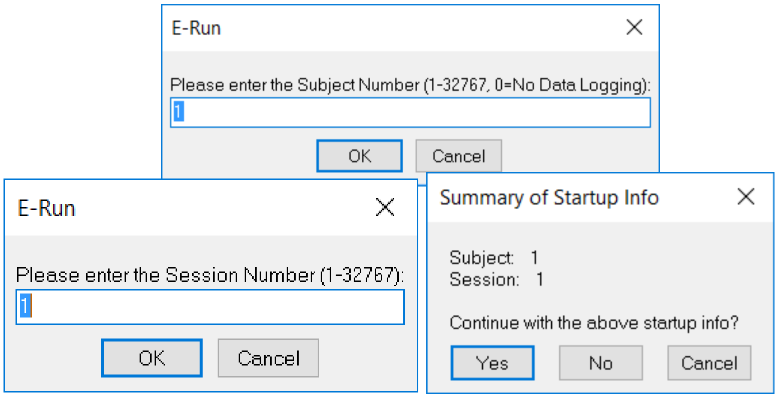
Run the experiment to verify the desired objects show up on the correct displays.

You have now completed the basic steps necessary to create an EET-enabled paradigm with dual monitor functionality. Before you run the task, be sure to hook up a second monitor. EET-enabled experiments can be run locally from E-Studio during development and testing. You should always fully test your experiment prior to scheduling actual participants or before using it to collect data.

1) ***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) ***Press*** **Enter** to ***accept*** the **default values** for each of the initialization prompts presented.



E-Run

Please enter the Subject Number (1-32767, 0=No Data Logging):

1

E-Run

Please enter the Session Number (1-32767):

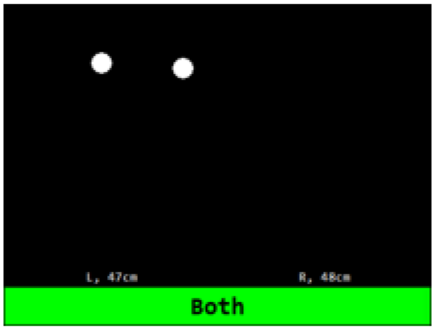
1

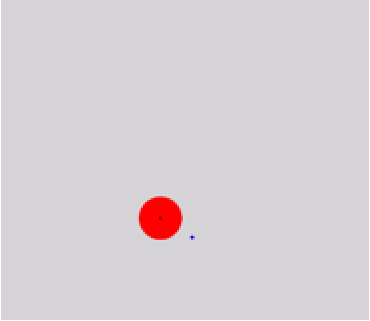
Summary of Startup Info

Subject: 1
Session: 1

Continue with the above startup info?

4) ***Look*** at the **participant screen** to ***verify*** that the **eyes** are **stable** in the Track Status window. ***Ensure*** that there are **two white dots** which appear in the box and that the **bottom bar** is **green**. ***Press*** **Space** once both eyes are stable.





5) ***Accept*** the **calibration (if acceptable)** and ***perform*** the **experiment**.

6) ***Observe*** the **stimulus presentation sequence** to ***verify*** the **experiment** is ***functioning*** correctly and ***completes*** with **no errors** being generated.

3.6. Tutorial 4: Working with Eye Gaze Data Interactively

Summary:

In *Tutorial 2: Logging Gaze Data in an Experiment*, eye gaze data was "passively" collected by E-Prime and combined with E-Prime data to create an output file. When used "actively", eye gaze data may be used to allow the paradigm to "interact" with the raw eye gaze data properties during the trial (as is the case with the routine "WaitForFixation").

E-Prime can be used to perform calculations for analysis at runtime. Runtime analysis results can be passed on to the data file for use in post hoc analysis. In this tutorial, eye gaze data is analyzed while it is collected to determine which object on the screen is fixated on by the participant.

This tutorial reviews the TETVaryingPositionAOITracking.es3 Sample experiment. Script is explained that actively accesses the gaze data during the trials. Due to the more complex nature of the tutorial, the experiment provided has incorporated all the necessary changes.

NOTE: *This tutorial does not have an accompanying file in the …My Experiments\3.0\Tobii Pro\Tutorials folder since the experiment provided has incorporated all the necessary changes. The experiment used in this tutorial is from the Samples folder.*

In TETVaryingPositionAOITracking.es3 a target is shown that needs to be identified later. The Target consists of a letter with a specific orientation (e.g., an upside-down A). A fixation is then shown followed by a randomly generated screen of the same letter in different orientations. The objective is to identify the original target and to locate that exact letter and orientation. As a participant looks through the letters, each becomes surrounded by a red box.

NOTE*: If not familiar with the* ***TETVaryingPositionAOITracking.es3*** *experiment, we recommend running the experiment now.*

Goal*:*

This tutorial teaches the concepts behind interacting with gaze data during the trials.

Overview of Tasks:

- Open TETVaryingPositionAOITracking.es3.
- Review the Stimulus Slide Object's Property Pages.
- Review the AOITracking InLine.

Recommended Readings:

For further information on the script discussed during this tutorial, please refer to the *E-Prime Command Reference*.

Estimated Tutorial Time*:*

10-15 minutes

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

Locate the E-Studio icon in the Start > All Programs > E-Prime menu and launch the application by selecting it. Load the TETVaryingPositionAOITracking.es3 sample experiment.

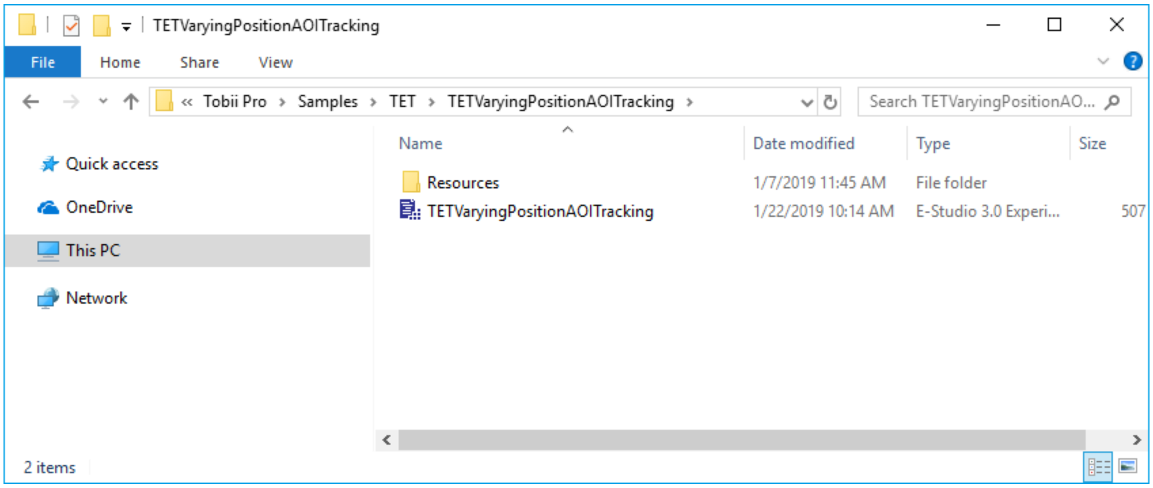
Open the E-Studio application, navigate to …My Experiments\3.0\Tobii Pro\Samples\TET\ TETVaryingPositionAOITracking, and load the TETVaryingPositionAOITracking.es3 sample experiment.

1) ***Click*** on the **Windows Start menu**, ***search*** for **E-Prime 3.0**. From the menu, ***click*** on **E-Studio** to launch the application.

2) ***Click*** the **Cancel** button on the New E-Prime 3.0 Experiment window.

3) ***Select*** **Open** from the **File** menu**.** ***Navigate*** to the **…My Experiments\3.0\Tobii Pro\Samples\TET\TETVaryingPositionAOITracking** folder to load the experiment.

4) ***Select*** the **"TETVaryingPositionAOITracking.es3"** file and then ***click*** the **Open** button to load the experiment into E-Studio.


TETVaryingPositionAOITracking
File
Home
Share
View
« Tobii Pro › Samples › TET › TETVaryingPositionAOITracking ›
Search TETVaryingPositionAO...
Quick access
OneDrive
This PC
Network
2 items

Name	Date modified	Type	Size
Resources	1/7/2019 11:45 AM	File folder	
TETVaryingPositionAOITracking	1/22/2019 10:14 AM	E-Studio 3.0 Experi...	507

NOTE: *If Samples and Tutorials for E-Prime Extensions for Tobii Pro 3.2 do not exist, see Troubleshooting, 4.1 Cannot locate E-Prime Samples and Tutorials.*

Task 2: Verify the Stimulus Slide Object Property Pages

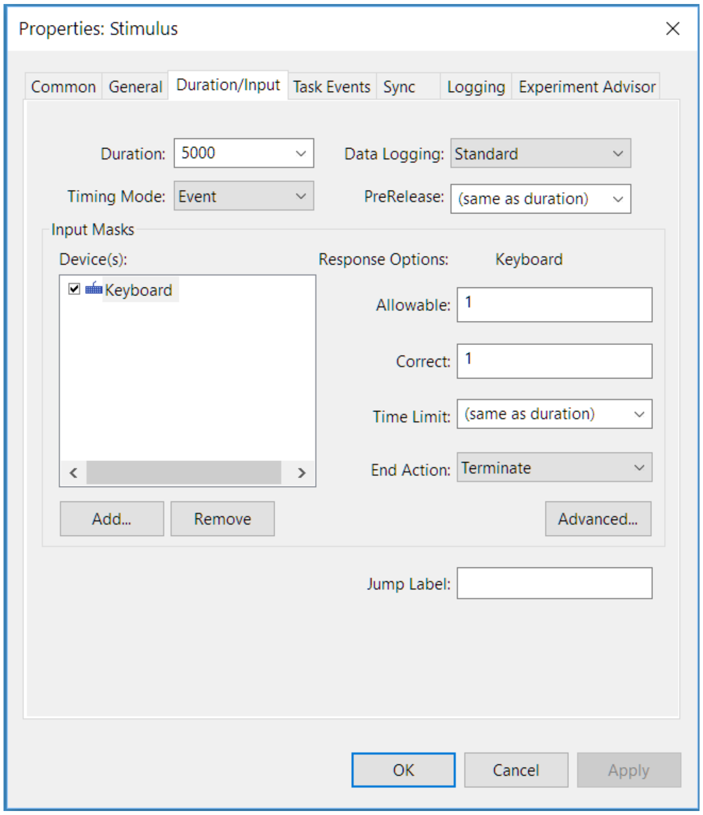
Open the Stimulus Slide Object Property Pages and verify the PreRelease and Duration Properties.

This experiment tracks the participant's gaze on screen in real time. To track the participant's gaze in real time, we need to confirm the **PreRelease** of the Stimulus Slide object to be the same as the duration. PreRelease controls the amount of time released during the processing of the current object to allow for setup of the next object. This allows resources that are being used for the Stimulus Slide object be released and made available for the AOITracking InLine. This gives the AOITracking InLine the ability to dynamically alter the border color of the AOI that is being viewed by the participant. For more information about PreRelease refer to *Article 22857: Stimulus Preparation Solutions*.

1) ***Double click*** the **Stimulus Slide** object to ***open*** it in the **workspace**.

2) ***Click*** the **Property Pages** button.

3) ***Select*** the **Duration/Input** tab.



Duration: 5000

Timing Mode: Event

Data Logging: Standard

PreRelease: (same as duration)

Input Masks

Device(s): ☑ Keyboard

Response Options: Keyboard

Allowable: 1

Correct: 1

Time Limit: (same as duration)

End Action: Terminate

Jump Label:

4) ***Confirm*** the **Duration** property is set to **5000**.

5) ***Confirm*** the **PreRelease** property is set to **(same as duration)**.

6) ***Confirm*** the **Allowable** and **Correct** properties are both **1**.

7) ***Click*** the **OK** button to accept the changes.

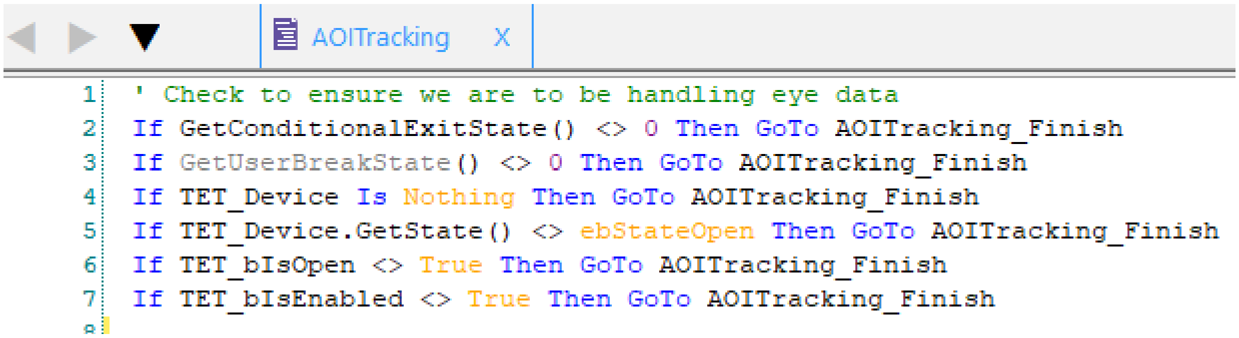
Task 3: Review the Script that ensures Eye Tracking is occurring

Review the script that handles eye data.

When testing an experiment, sometimes it is not necessary to run with an eye tracker (refer to *Troubleshooting, 4.2 Running without an eye tracker*). When implementing the AOI Tracking script, the first lines of the InLine handle the instance where a researcher would run without an eye tracker or TET capabilities. If an eye tracker is not being used or if a researcher specifies a break in the experiment, the script reviewed in this Task jumps to a Label object at the end of the InLine script. Jumping to the Label object effectively “skips” the AOI tracking script between the Label and GoTo script.

1) ***Double click*** the **AOITracking InLine** to open in the workspace.

2) ***View* lines 1-7** in the InLine Script.



```
1  ' Check to ensure we are to be handling eye data
2  If GetConditionalExitState() <> 0 Then GoTo AOITracking_Finish
3  If GetUserBreakState() <> 0 Then GoTo AOITracking_Finish
4  If TET_Device Is Nothing Then GoTo AOITracking_Finish
5  If TET_Device.GetState() <> ebStateOpen Then GoTo AOITracking_Finish
6  If TET_bIsOpen <> True Then GoTo AOITracking_Finish
7  If TET_bIsEnabled <> True Then GoTo AOITracking_Finish
8
```

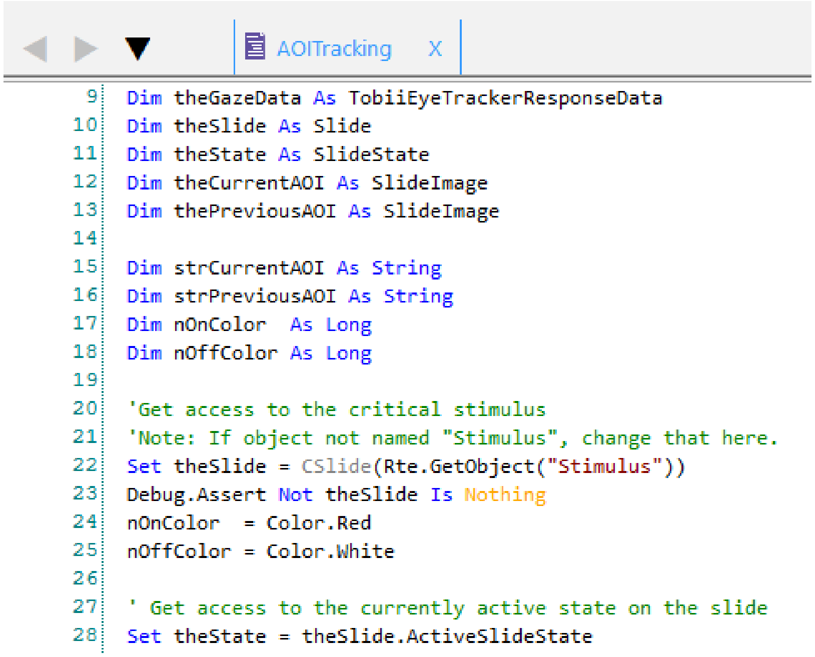
Script	Purpose
`If GetConditionalExitState() <> 0 Then GoTo AOITracking_Finish`	Jumps to AOITracking_Finish Label if a Conditional Exit occurs.
`If GetUserBreakState() <> 0 Then GoTo AOITracking_Finish`	Jumps to AOITracking_Finish Label if the UserBreakState occurs.
`If TET_Device Is Nothing Then GoTo AOITracking_Finish`	Jumps to AOITracking_Finish Label if the TET Device is not set.
`If TET_Device.GetState() <> ebStateOpen Then GoTo AOITracking_Finish`	Jumps to AOITracking_Finish Label if the TET Device is not at an opened state.
`If TET_bIsOpen <> True Then GoTo AOITracking_Finish`	Jumps to AOITracking_Finish Label if the TET Package is not at an opened state.
`If TET_bIsEnabled <> True Then GoTo AOITracking_Finish`	Jumps to AOITracking_Finish Label if the TET Package is disabled

Task 4: Review the Variable Declarations

Verify variable declarations.

Before examining the Do…Loop that contains the AOI tracking script, it is necessary to Declare and set some variables. By using the **Dim** statement, variables and their associated types and sizes can be declared. The **Set** statement allows users to assign objects to object variables.

1) ***View* lines 9-28** in the InLine Script.



```
 9  Dim theGazeData As TobiiEyeTrackerResponseData
10  Dim theSlide As Slide
11  Dim theState As SlideState
12  Dim theCurrentAOI As SlideImage
13  Dim thePreviousAOI As SlideImage
14
15  Dim strCurrentAOI As String
16  Dim strPreviousAOI As String
17  Dim nOnColor  As Long
18  Dim nOffColor As Long
19
20  'Get access to the critical stimulus
21  'Note: If object not named "Stimulus", change that here.
22  Set theSlide = CSlide(Rte.GetObject("Stimulus"))
23  Debug.Assert Not theSlide Is Nothing
24  nOnColor  = Color.Red
25  nOffColor = Color.White
26
27  ' Get access to the currently active state on the slide
28  Set theState = theSlide.ActiveSlideState
```

Script	Purpose
`Dim theGazeData As TobiiEyeTrackerResponseData`	Declares variable to store the gaze data sent from the Tobii Pro Eye Tracker.
`Dim theSlide As Slide`	Declares variable for the Slide.
`Dim theState As SlideState`	Declares variable for the Slide state.
`Dim theCurrentAOI As SlideImage`	Declares variable for the Slide image sub-object which represents the AOI that is currently being viewed.
`Dim thePreviousAOI As SlideImage`	Declares variable for the Slide image sub-object which represents the AOI that was previously viewed.
`Dim strCurrentAOI As String`	Declares variable that saves the current AOI data.
`Dim strPreviousAOI As String`	Declares variable that saves the previous AOI data.
`Dim nOnColor As Long`	Declares variable that sets the “on” color for the border.
`Dim nOffColor As Long`	Declares variable that sets the “off” color for the border.
`Set theSlide = CSlide(Rte.GetObject("Stimulus"))`	Sets theSlide variable to be the Stimulus Slide. If the Slide has a different name, then value in quotes needs to be changed to reflect the name of the Slide.
`Debug.Assert Not theSlide Is Nothing`	Generates a fatal error if the expression evaluates to false.
`nOnColor  = Color.Red`	Sets the “on” color to red when viewing the AOI.
`nOffColor = Color.White`	Sets the “off” color to white when not viewing the AOI.
`Set theState = theSlide.ActiveSlideState`	Sets theState to be the active State (i.e., State1) of the Slide (i.e., Stimulus).

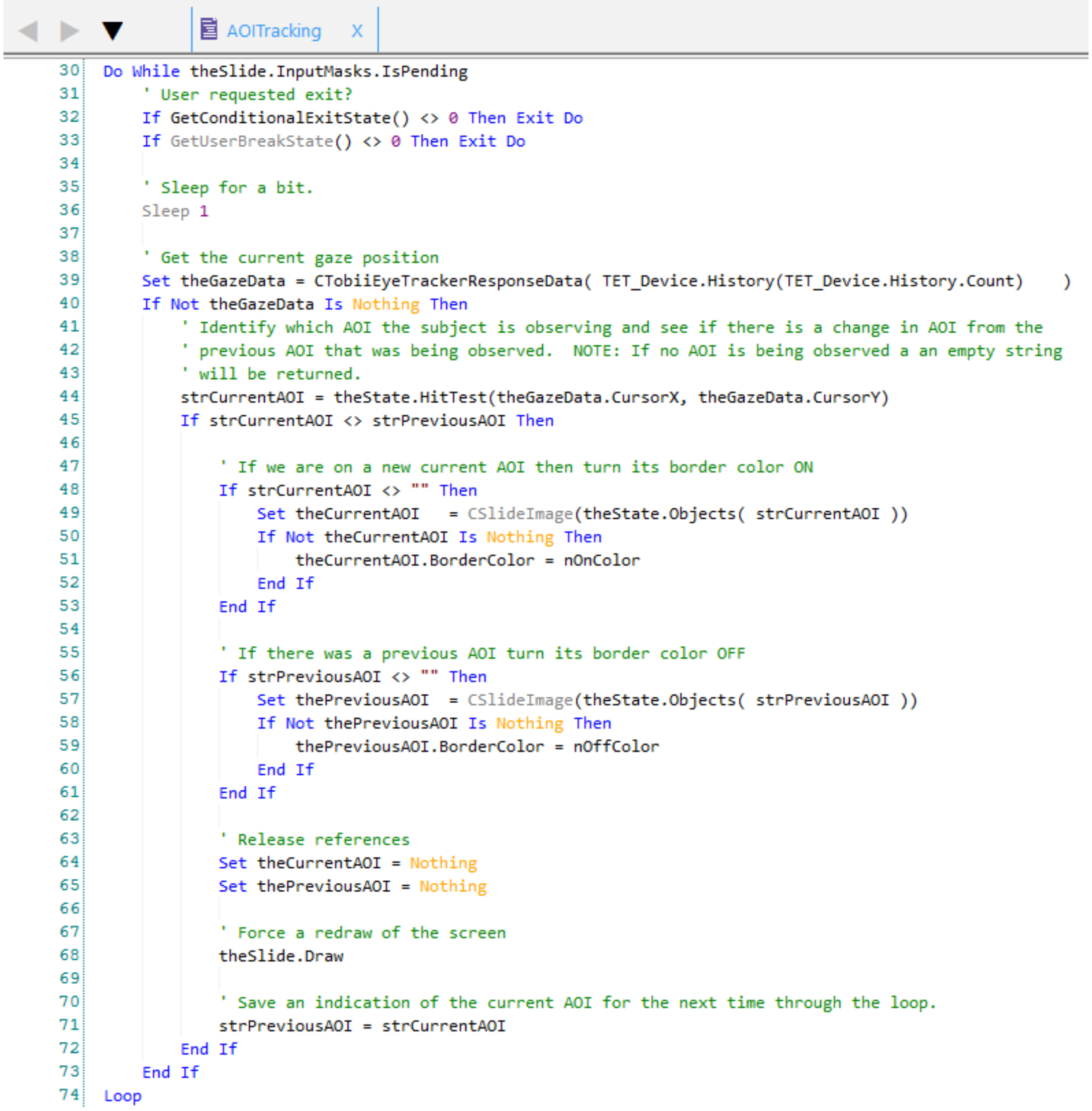
Task 5: Review the Do…Loop that performs AOI Tracking

Understand the basic function of the Do…Loop and how it works in the AOITracking InLine.

A **Do…Loop** is used in this experiment in conjunction with the theSlide.InputMasks.IsPending method to continuously evaluate which, if any, AOI is being fixated on until a response is made. This allows E-Prime to analyze the eye gaze data to determine when a fixation moves “off” and “on” the available AOIs. This information is used to provide feedback to the participant for what object they are fixating on.

A **HitTest** is performed in the Do…Loop using theGazeData.CursorX and CursorY positions. The Hit Test is used to determine which AOI is currently being viewed by a participant. After that is determined, the appearance of the Slide border is altered to show the participant which letter is being gazed upon.

1) ***View*** **lines 30-74** in the InLine Script.



```
30 Do While theSlide.InputMasks.IsPending
31     ' User requested exit?
32     If GetConditionalExitState() <> 0 Then Exit Do
33     If GetUserBreakState() <> 0 Then Exit Do
34 
35     ' Sleep for a bit.
36     Sleep 1
37 
38     ' Get the current gaze position
39     Set theGazeData = CTobiiEyeTrackerResponseData( TET_Device.History(TET_Device.History.Count)    )
40     If Not theGazeData Is Nothing Then
41         ' Identify which AOI the subject is observing and see if there is a change in AOI from the
42         ' previous AOI that was being observed.  NOTE: If no AOI is being observed a an empty string
43         ' will be returned.
44         strCurrentAOI = theState.HitTest(theGazeData.CursorX, theGazeData.CursorY)
45         If strCurrentAOI <> strPreviousAOI Then
46 
47             ' If we are on a new current AOI then turn its border color ON
48             If strCurrentAOI <> "" Then
49                 Set theCurrentAOI   = CSlideImage(theState.Objects( strCurrentAOI ))
50                 If Not theCurrentAOI Is Nothing Then
51                     theCurrentAOI.BorderColor = nOnColor
52                 End If
53             End If
54 
55             ' If there was a previous AOI turn its border color OFF
56             If strPreviousAOI <> "" Then
57                 Set thePreviousAOI  = CSlideImage(theState.Objects( strPreviousAOI ))
58                 If Not thePreviousAOI Is Nothing Then
59                     thePreviousAOI.BorderColor = nOffColor
60                 End If
61             End If
62 
63             ' Release references
64             Set theCurrentAOI = Nothing
65             Set thePreviousAOI = Nothing
66 
67             ' Force a redraw of the screen
68             theSlide.Draw
69 
70             ' Save an indication of the current AOI for the next time through the loop.
71             strPreviousAOI = strCurrentAOI
72         End If
73     End If
74 Loop
```

Script	Purpose
Do While theSlide.InputMasks.IsPending	Loops the script while the Slide Object Input Mask is armed and awaiting input or disarmed and therefore not accepting input. It has a corresponding Loop statement.
If GetConditionalExitState() <> 0 Then Exit Do	Exits the Loop if a Conditional Exit occurs.
If GetUserBreakState() <> 0 Then Exit Do	Exits the Loop if the UserBreakState occurs.
Sleep 1	Sleeps for a millisecond during the Loop.
Set theGazeData = CTobiiEyeTrackerResponseData(TET_Device.History(TET_Device.History.Count))	Sets the gaze data to be the Count from the eye tracker history.
If Not theGazeData Is Nothing Then	Performs the conditional if theGazeData is set to something. It has a corresponding End If statement.
strCurrentAOI = theState.HitTest(theGazeData.CursorX, theGazeData.CursorY)	Performs a HitTest on the strCurrentAOI.
If strCurrentAOI <> strPreviousAOI Then	Uses a conditional statement to determine if the current AOI is the same as the previous AOI. It has a corresponding End If statement.

If strCurrentAOI <> "" Then
 Set theCurrentAOI=CSlideImage(theState.Objects(strCurrentAOI))
  If Not theCurrentAOI Is Nothing Then
   theCurrentAOI.BorderColor = nOnColor
  End If
End If
	Use a conditional statement to switch on the BorderColor if the AOI is currently being viewed.

If strPreviousAOI <> "" Then
 Set
thePreviousAOI=CSlideImage(theState.Objects(strPreviousAOI))
  If Not thePreviousAOI Is Nothing Then
   thePreviousAOI.BorderColor = nOffColor
  End If
End If
	Uses a conditional statement to switch off the BorderColor if the AOI is not currently being viewed.

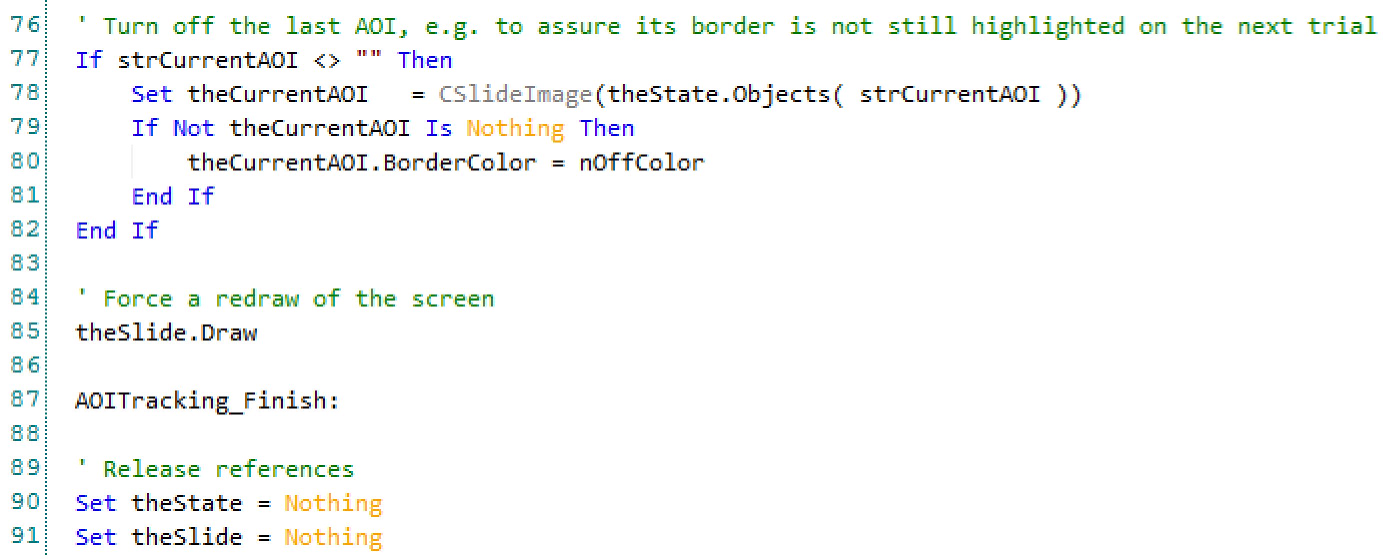
Set theCurrentAOI = Nothing Set thePreviousAOI = Nothing	Releases theCurrentAOI and thePreviousAOI references.
theSlide.Draw	Redraws the screen.
strPreviousAOI = strCurrentAOI	Saves the current AOI for the next time through the loop.

Task 6: Review the Values set for the Next Trial

Clears the references and redraws the border for the next trial.

Before the next trial commences the values need to be reset. In this part of the AOITracking InLine, the variables are reset to turn off the BorderColor; the screen is redrawn, and finally the variable references are released.

1) ***View* lines 76-91** in the InLine Script.



```
76  ' Turn off the last AOI, e.g. to assure its border is not still highlighted on the next trial
77  If strCurrentAOI <> "" Then
78      Set theCurrentAOI   = CSlideImage(theState.Objects( strCurrentAOI ))
79      If Not theCurrentAOI Is Nothing Then
80          theCurrentAOI.BorderColor = nOffColor
81      End If
82  End If
83
84  ' Force a redraw of the screen
85  theSlide.Draw
86
87  AOITracking_Finish:
88
89  ' Release references
90  Set theState = Nothing
91  Set theSlide = Nothing
```

Script	Purpose

`If strCurrentAOI <> "" Then`
`Set theCurrentAOI = CSlideImage(theState.Objects(strCurrentAOI))`
`If Not theCurrentAOI Is Nothing Then`
`theCurrentAOI.BorderColor = nOffColor`
`End If`
`End If`
	Turns off the BorderColor.
	This ensures any AOIs are not highlighted at the beginning of the next trial.

`theSlide.Draw`	Redraws the screen.
`AOITracking_Finish:`	Indicates the end of the AOITracking InLine. Acts like a Label Object.

`Set theState = Nothing`
`Set theSlide = Nothing`
	Releases theState and theSlide references.

3.7. Tutorial 5: Adding Tobii Pro Lab Support to an Experiment

Summary:

Incorporating Tobii Pro Lab Support into an E-Prime experiment involves adding the TobiiProLab PackageFile and the TobiiProLab PackageCalls to an experimental structure at the appropriate locations. The required parameters may then need edited to allow the TobiiProLab PackageCall to meet an experiment’s needs.

During this tutorial, Tobii Pro Lab support is added to the TETFixedPositionAOI.es3 sample experiment edited in Tutorial 1. Support is added that allows E-Prime to start and stop a recording in Tobii Pro Lab. Additionally, PackageCalls are added that send display information to Tobii Pro Lab.

PackageCalls are added during the experiment that send Areas of Interest (AOIs) to Tobii Pro Lab. The AOIs are used by researchers when analyzing stimuli that participants frequently view during an experiment. Furthermore, Tags can be used to help categorize and identify information about AOIs. This tutorial shows how to add both AOI and Tag support to a Tobii Pro Lab-enabled experiment.

Using the eye tracker, Tobii Pro Lab records where a participant’s eyes were looking during an experiment. This recording can be used to produce heat maps, show eye tracking visualizations, and calculate advanced metrics in Tobii Pro Lab.

NOTE: *The sample experiment found in the folder …My Experiments\3.0\Tobii Pro\Samples\TPL\TPLFixedPositionAOI represents the complete set of changes documented in this tutorial.*

***Goal**:*

This tutorial illustrates how to add the Tobii Pro Lab PackageCalls into the FixedPositionAOI.es3 tutorial experiment. When this tutorial is completed, a “Tobii Pro Lab-enabled” paradigm is completed.

Overview of Tasks:

- Open and save TETFixedPositionAOI.es3.
- Add the TobiiProLab PackageFile to the Experiment Object.
- Add the TobiiProLab to the Devices tab.
- Edit the TobiiProLab device.
- Add the PackageCalls to open Tobii Pro Lab communication.
- Add the PackageCalls to start recording in Tobii Pro Lab.
- Add the PackageCalls to stop recording in Tobii Pro Lab.
- Add the PackageCalls to close Tobii Pro Lab communication.
- Add the PackageCall to set and send stimuli display events.
- Add the PackageCall to send AOIs to Tobii Pro Lab.
- Add the PackageCall to send Tags to Tobii Pro Lab.
- Add the PackageCall to signify completion of the Stimulus Display events.
- Prepare the Tobii Pro Lab software.
- Verify the overall experiment structure and run the experiment.
- Verify the recording, AOIs, and tags were sent to Tobii Pro Lab.

Recommended readings:
Prior to beginning this tutorial, we recommend reading the Tobii Pro Lab User Manual to gain an understanding of how to use Tobii Pro Lab.

Estimated Tutorial Time:
15-20 minutes

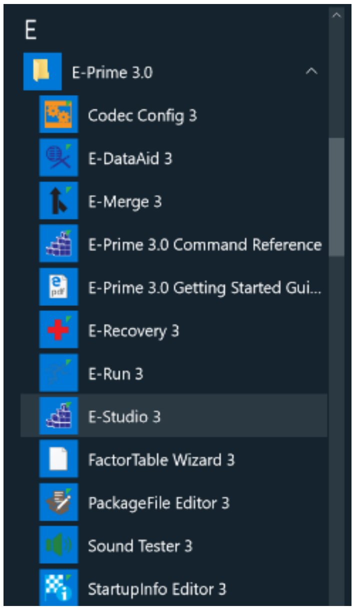
Task 1: Open the FixedPositionAOI.es3 Experiment in E-Studio

Locate the E-Studio icon in the Start > All Programs > E-Prime 3.0 menu and launch the application by selecting it. Load the FixedPositionAOI.es3 tutorial experiment.

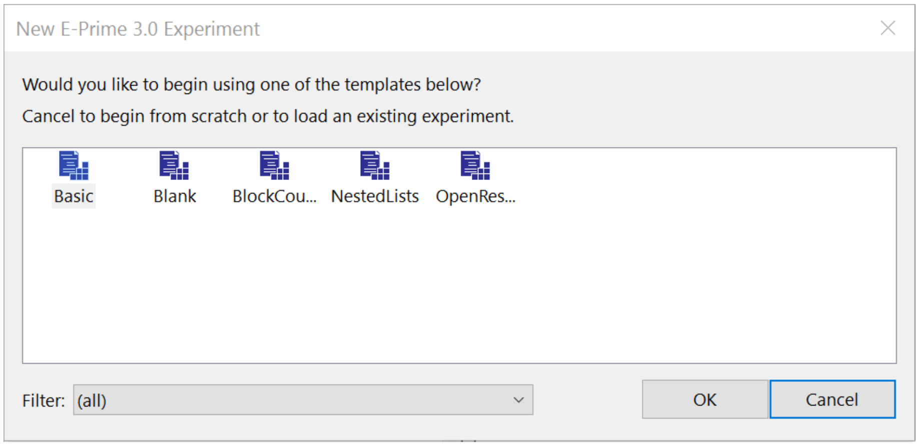
The **E-Studio** application is installed as part of the typical E-Prime installation. This application is used to create, modify, and test experiments within E-Prime. Open the E-Studio application, navigate to …My Experiments\3.0\Tobii Pro\Tutorials\TPL\TPLFixedPositionAOI, and load the TETFixedPositionAOI.es3 tutorial experiment.

1) ***Click*** on the **Windows Start** menu, ***search*** for **E-Prime 3.0**.

2) From the menu, ***click*** on **E-Studio** to launch the application.



3) ***Click*** the **Cancel** button on the New E-Prime 3.0 Experiment window.


New E-Prime 3.0 Experiment

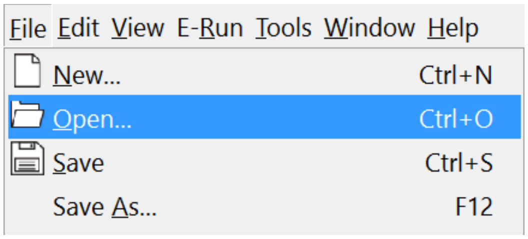
Would you like to begin using one of the templates below?

Cancel to begin from scratch or to load an existing experiment.

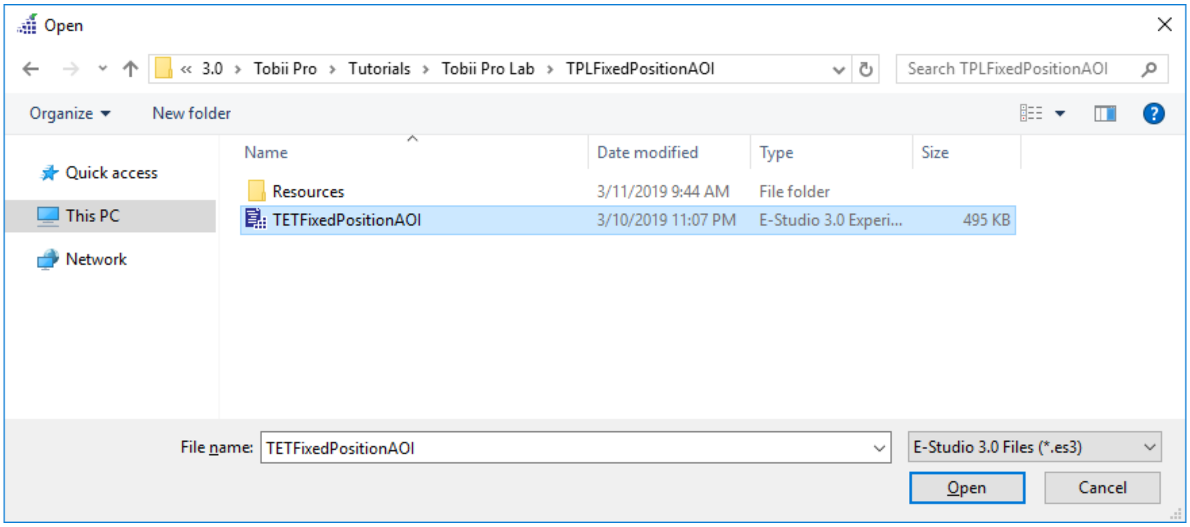
Filter: (all)

4) ***Select*** **Open** from the **File** menu**.**

5) ***Navigate*** to the …**My Experiments\3.0\TobiiPro\Tutorials\TPL\TPLFixedPositionAOI** folder to load the experiment.



6) ***Select*** the **"TETFixedPositionAOI.es3"** file and then ***click*** the **Open** button to load the experiment into E-Studio.


Open
« 3.0 › Tobii Pro › Tutorials › Tobii Pro Lab › TPLFixedPositionAOI
Search TPLFixedPositionAOI
Organize ▾
New folder
Quick access
This PC
Network
Open
Cancel

Name	Date modified	Type	Size
Resources	3/11/2019 9:44 AM	File folder	
TETFixedPositionAOI	3/10/2019 11:07 PM	E-Studio 3.0 Experi...	495 KB

File name: TETFixedPositionAOI

E-Studio 3.0 Files (*.es3)

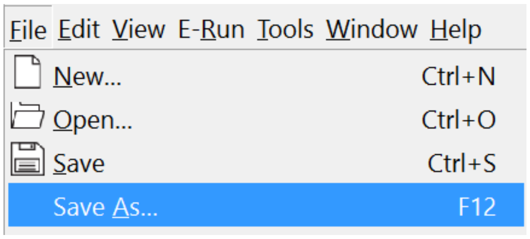
NOTE: *If Samples and Tutorials for E-Prime Extensions for Tobii Pro 3.2 do not exist, see Troubleshooting, 4.1 Cannot locate E-Prime Samples and Tutorials.*

Task 2: Save the experiment under a new name

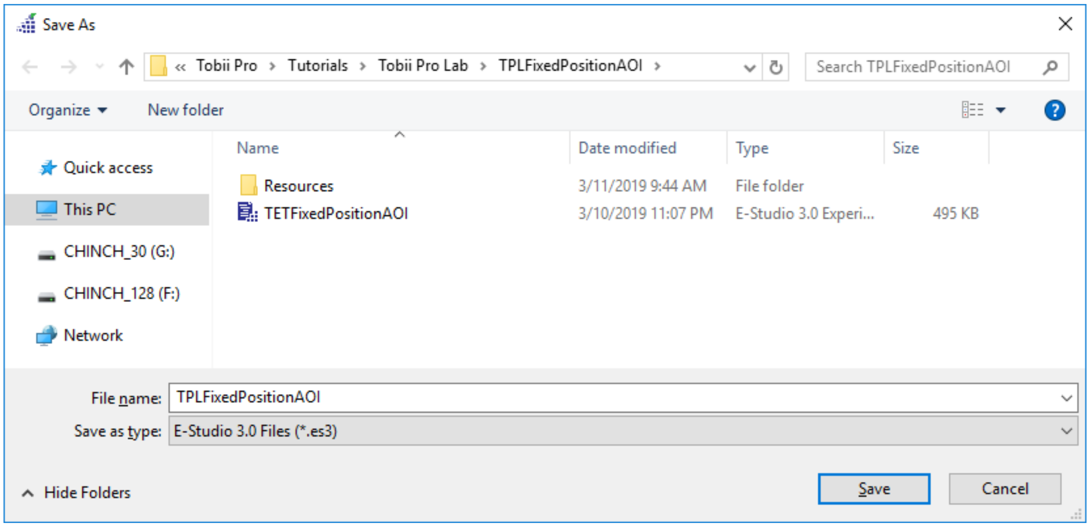
Save the FixedPositionAOI.es3 experiment in the same folder under a new name.

Rename the experiment but be sure to save it in the same folder (…My Experiments\3.0\Tobii Pro\Tutorials\TPL\TPLFixedPositionAOI) so that any resources within the experiment will remain valid and can be reused.

1) ***Select*** **Save As...** from the **File** menu**.**



2) ***Type*** "**TPLFixedPositionAOI.es3**" as the new name in the **File name** field. ***Click*** the **Save** button.



Name	Date modified	Type	Size
Resources	3/11/2019 9:44 AM	File folder	
TETFixedPositionAOI	3/10/2019 11:07 PM	E-Studio 3.0 Experi...	495 KB

File name: TPLFixedPositionAOI

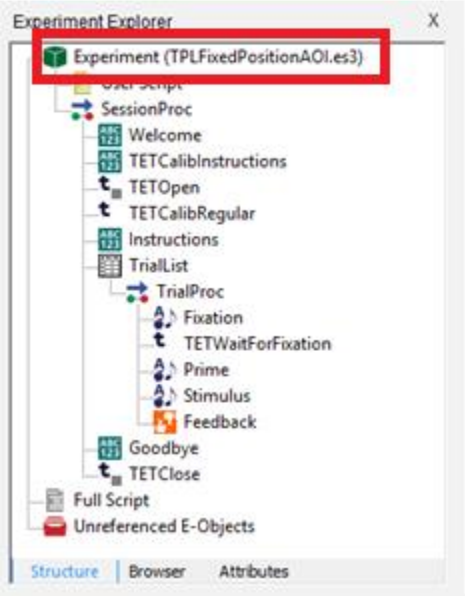
Save as type: E-Studio 3.0 Files (*.es3)

Task 3: Add the TobiiProLab Package to the Experiment Property Pages

Open the Properties dialog for the Experiment Object and use the Packages tab to add the TobiiProLab PackageFile to the experiment.

PackageFiles in E-Prime are cohesive sets of E-Basic routines that are grouped together. To gain access to the routines within a PackageFile you must first add the PackageFile to the experiment. PackageFiles can be added to an experiment using the Packages tab of the Experiment Object Properties dialog. The routines that are used to communicate with Tobii Pro Lab at runtime are contained within the TobiiProLab PackageFile. The PackageFile also includes a customized Toolbox which contains shortcuts that you can drag into Procedure Objects within your experiment to invoke routines contained in the PackageFile.

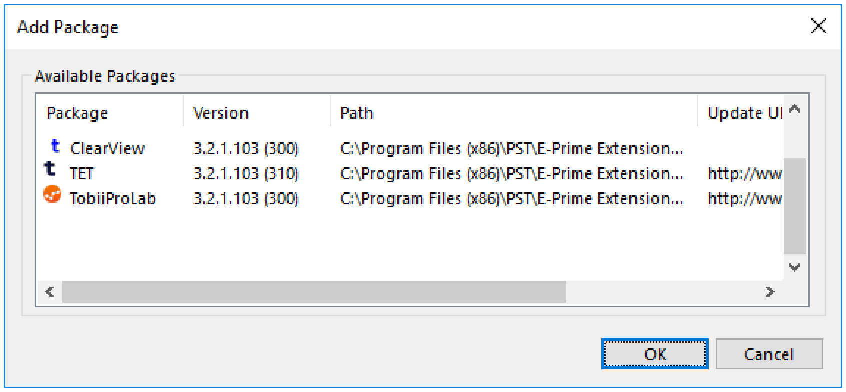
1) ***Double click*** the **Experiment Object** at the top of the tree in the **Experiment Explorer** window.



2) ***Click*** on the **Packages** tab of the **Experiment Object Properties** dialog.

3) ***Click*** the **Add...** button.

4) ***Select*** the **TobiiProLab PackageFile** in the **Add Package** dialog.

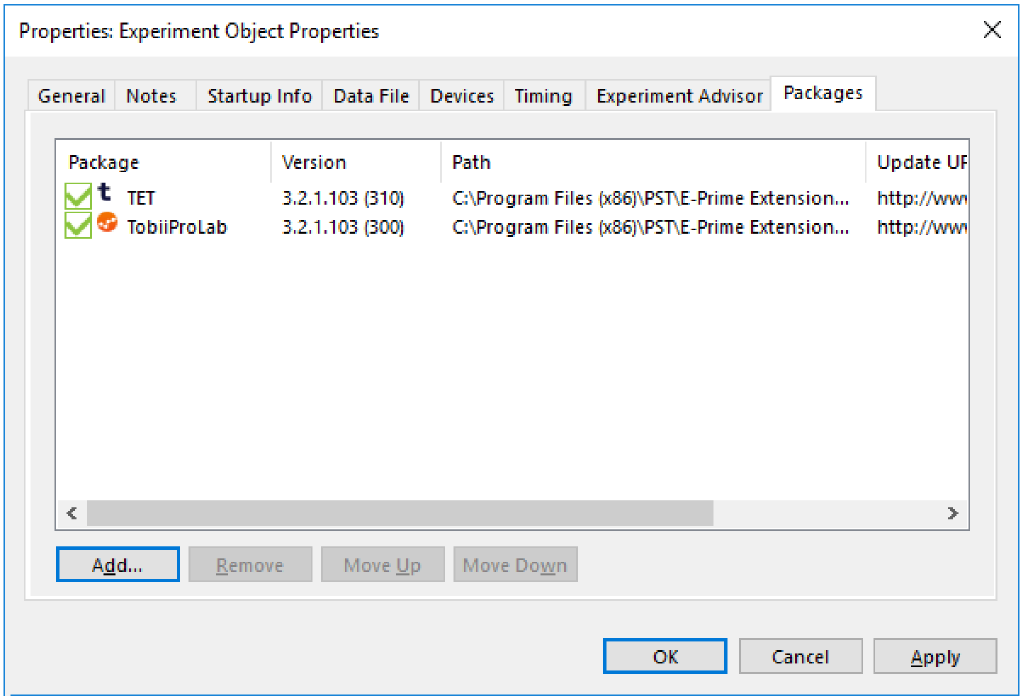


Package	Version	Path	Update UI
ClearView	3.2.1.103 (300)	C:\Program Files (x86)\PST\E-Prime Extension...	
TET	3.2.1.103 (310)	C:\Program Files (x86)\PST\E-Prime Extension...	http://ww
TobiiProLab	3.2.1.103 (300)	C:\Program Files (x86)\PST\E-Prime Extension...	http://ww

NOTE: *The TET PackageFile is already added to the experiment by default.*

5) ***Click*** the **OK** button to accept the changes.

6) ***Verify*** the **TobiiProLab PackageFile** has been added and is checked (<u>**Do not**</u> yet ***click*** **OK** or ***dismiss*** the dialog).



Package	Version	Path	Update UF
☑ TET	3.2.1.103 (310)	C:\Program Files (x86)\PST\E-Prime Extension...	http://www
☑ TobiiProLab	3.2.1.103 (300)	C:\Program Files (x86)\PST\E-Prime Extension...	http://www

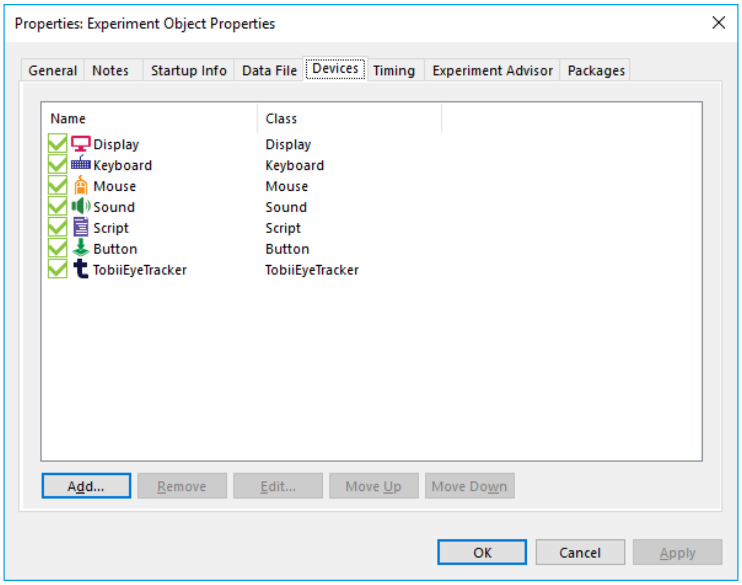
NOTE: *The PackageFile version number that is displayed by E-Studio reflects the version of the TobiiProLab PackageFile that is currently installed on your machine and may not match the picture.*

Task 4: Add the TobiiProLab Device to the Experiment Properties

Select the Devices tab to add the TobiiProLab Device to the experiment.

Select the Devices tab of the Experiment Object Property Pages to add the TobiiProLab device to the list of devices. Verify that it is the last device shown.

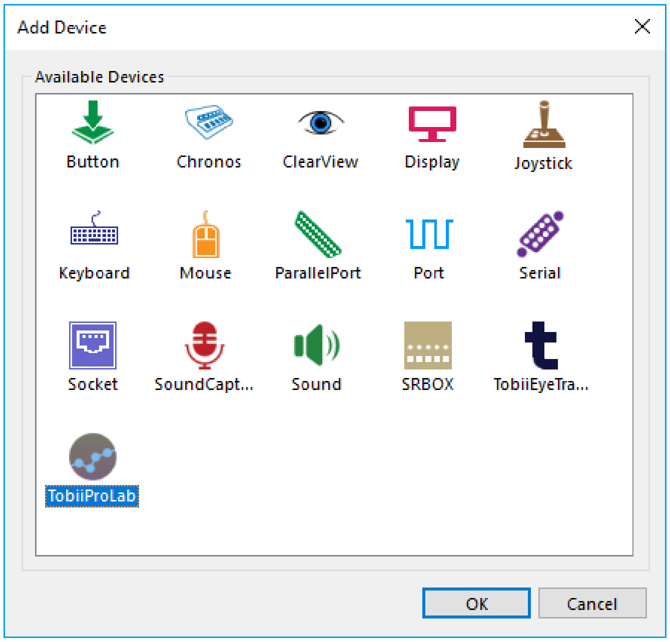
1) ***Click*** on the **Devices** tab of the **Experiment Object Properties** dialog.



Name	Class
☑ Display	Display
☑ Keyboard	Keyboard
☑ Mouse	Mouse
☑ Sound	Sound
☑ Script	Script
☑ Button	Button
☑ TobiiEyeTracker	TobiiEyeTracker

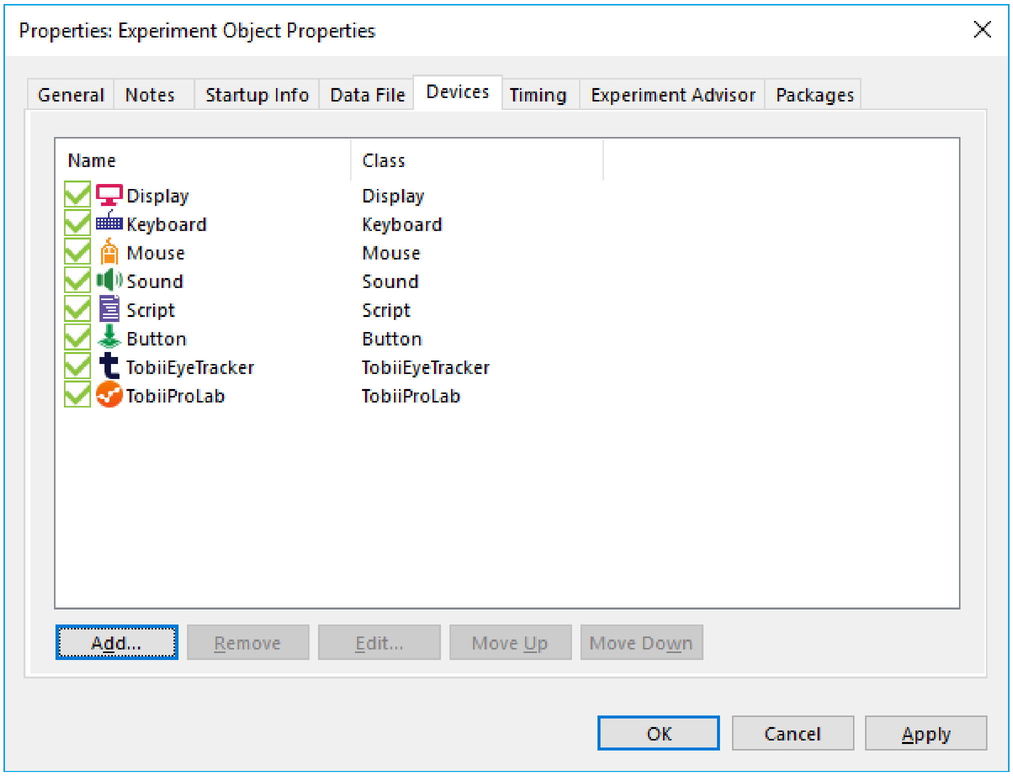
2) ***Click*** the **Add...** button.

3) ***Select*** the **TobiiProLab** Device in the **Add Device** dialog.



4) ***Click*** the **OK** button to accept the changes.

5) ***Verify*** the **TobiiProLab Device** is listed last under the **Name** column and is checked. ***Click*** the **OK** button to dismiss the **Add Device dialog**.



Name	Class
☑ Display	Display
☑ Keyboard	Keyboard
☑ Mouse	Mouse
☑ Sound	Sound
☑ Script	Script
☑ Button	Button
☑ TobiiEyeTracker	TobiiEyeTracker
☑ TobiiProLab	TobiiProLab

NOTE: *The order of devices from top to bottom is the order in which the devices become initialized in the beginning of an experimental run. Do not move the TobiiProLab Device to the top as the Display device must open prior to the TobiiProLab device. If this order is not followed, the experiment terminates.*

Task 5: Edit the TobiiProLab Device Properties

Open and edit the Property Pages for the TobiiProLab Device.

The **TobiiProLab** Device Properties can be configured to meet an eye tracker's or experiment's specific needs. Before running a TPL-enabled experiment, be sure to edit the TobiiEyeTracker Device Properties for the eye tracker used during runtime.

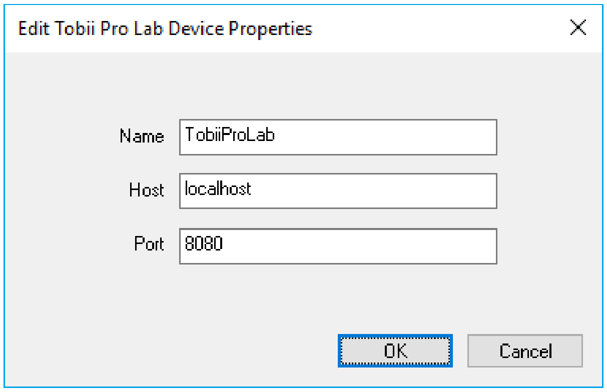
The first property (**Name**) allows for customization of the TobiiProLab device name. The default value for this property is TobiiProLab.

The **Host** property should be configured when running an experiment on a multiple machine setup. The IP Address of the machine running Tobii Pro Lab needs to be entered.
Refer to *Hardware Configuration with Two Machines* for more information on how to use a two-machine setup.

The **Port** property allows for specification of the Port number. The default value for this property is 8080.

1) ***Double click*** **TobiiProLab** to display the **TobiiProLab Device Properties** dialog.

2) ***Verify*** the **Name**, **Host**, and **Port** properties. ***Specify*** **appropriate values** for the experiment.



Edit Tobii Pro Lab Device Properties

Name: TobiiProLab
Host: localhost
Port: 8080

NOTE: *Refer to Appendix B: Hardware Configurations and Task 14: Prepare the Tobii Pro Lab Project for information on the Host and Port (depending on the machine configuration) and the Expected Project properties.*

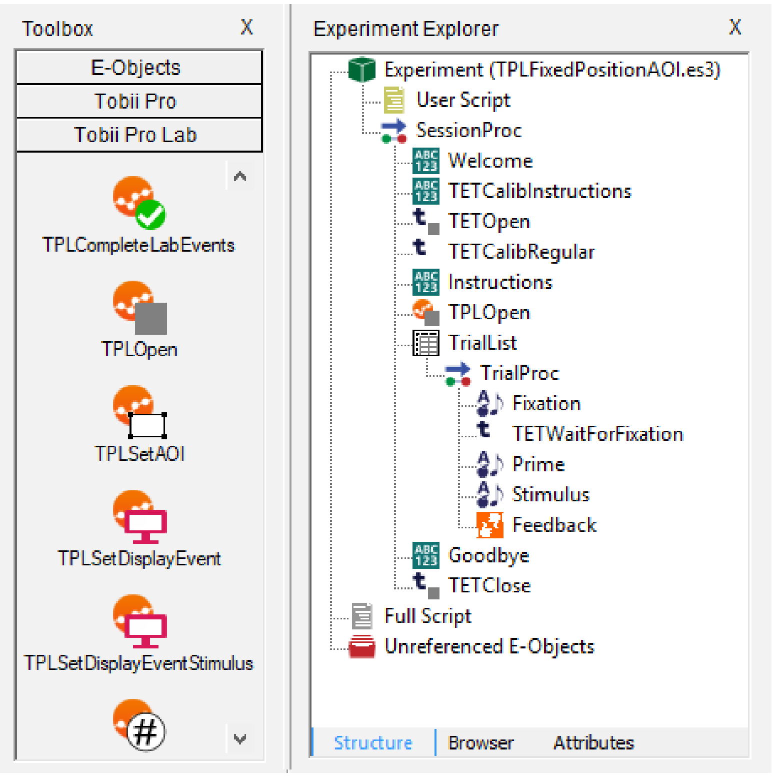
3) ***Click*** the **OK** button to accept the **TobiiProLab Device Properties dialog**. ***Click*** the **OK** button again to accept the **Experiment Object Properties dialog**.

Task 6: Add the TPLOpen PackageCall to Initialize Tobii Pro Lab Communication

Add a PackageCall to the SessionProc to enable communication with Tobii Pro Lab.

The **TPLOpen** PackageCall opens the TobiiProLab package and enables the routines to be executed after confirming that the TobiiProLab Device is enabled. This enables experiment information and eye tracking movements to be recorded to the project that is currently open in the Tobii Pro Lab application. It is preconfigured to call the Open routine. The TPLOpen PackageCall is typically added near the beginning of the SessionProc.

1) ***Drag*** the shortcut **TPLOpen** from the **Tobii Pro Lab Toolbox** and ***drop*** it in the **SessionProc after Instructions**.

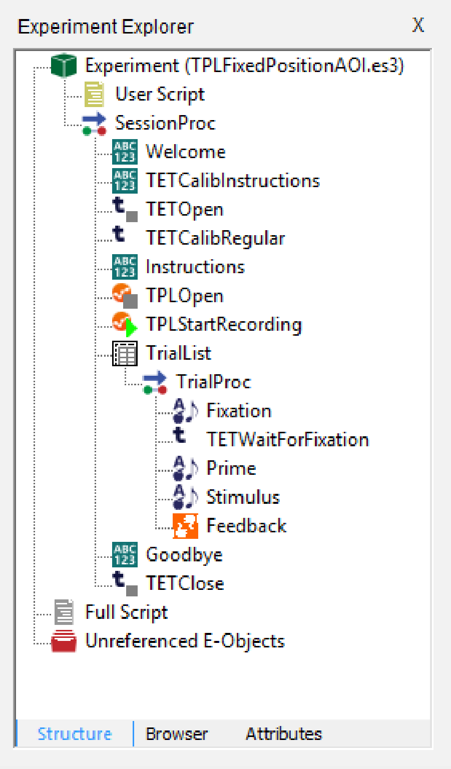


Task 7: Add the TPLStartRecording PackageCall to Begin Recording in Tobii Pro Lab

Add a PackageCall to the SessionProc to start the Tobii Pro Lab recording.

The next step is to add the **TPLStartRecording** PackageCall. This PackageCall is placed at the location in the procedure where the Tobii Pro Lab recording begins. This is often before the trials begin. Once a recording is started, it cannot be paused and resumed. A recording in Tobii Pro Lab can only be stopped. Additionally, multiple TPLStartRecordings cannot be called during the same experiment.

1) ***Drag*** the shortcut **TPLStartRecording** from the **Tobii Pro Lab Toolbox** and ***drop*** it in the **SessionProc after TPLOpen**.



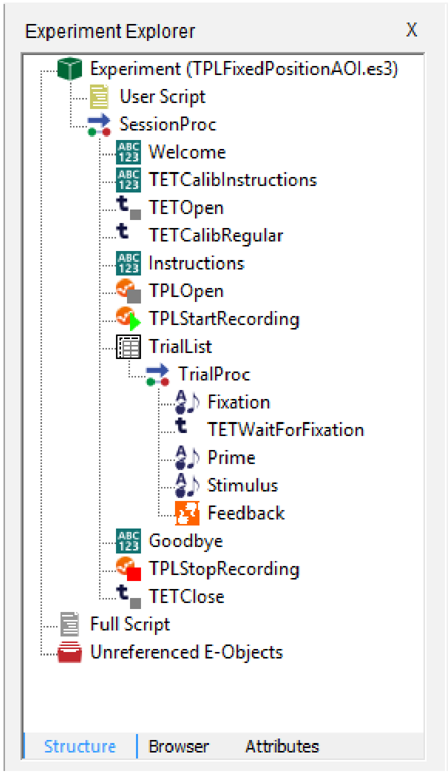
Task 8: Add the TPLStopRecording PackageCall to Stop Recording in Tobii Pro Lab

Add a PackageCall to the end of the SessionProc to stop the Tobii Pro Lab recording.

The **TPLStopRecording** PackageCall must be added to an experiment to stop the Tobii Pro Lab recording. This PackageCall is placed at the location in the procedure where the Tobii Pro Lab recording ends. This is often after all the trials have been completed.

NOTE: *Once a recording has been stopped, a recording cannot be started again in the same test run. Only one recording per test session may exist. Additionally, a recording cannot be paused.*

1) ***Drag*** the shortcut **TPLStopRecording** Object from the **Tobii Pro Lab Toolbox** and ***drop*** it after the **Goodbye** Object in the SessionProc.



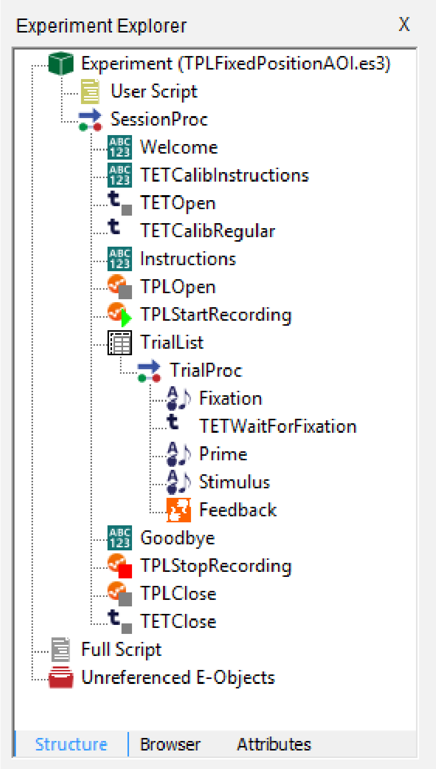
```
Experiment (TPLFixedPositionAOI.es3)
    User Script
    SessionProc
        Welcome
        TETCalibInstructions
        TETOpen
        TETCalibRegular
        Instructions
        TPLOpen
        TPLStartRecording
        TrialList
            TrialProc
                Fixation
                TETWaitForFixation
                Prime
                Stimulus
                Feedback
        Goodbye
        TPLStopRecording
        TETClose
Full Script
Unreferenced E-Objects
```

Task 9: Add the TPLClose PackageCall to Close Tobii Pro Lab Communication

Add a PackageCall to the SessionProc to disables communication with Tobii Pro Lab.

The last necessary PackageCall in any Tobii Pro Lab-enabled experiment is **TPLClose**. This PackageCall closes communication with Tobii Pro Lab. It should be implemented once communication to Tobii Pro Lab is no longer necessary in the experiment.

1) ***Drag*** the shortcut **TPLClose** from the **Tobii Pro Toolbox** and ***drop*** it after the **TPLStopRecording** Object in the **SessionProc.**



Experiment Explorer X

- Experiment (TPLFixedPositionAOI.es3)
  - User Script
  - SessionProc
    - Welcome
    - TETCalibInstructions
    - TETOpen
    - TETCalibRegular
    - Instructions
    - TPLOpen
    - TPLStartRecording
    - TrialList
      - TrialProc
        - Fixation
        - TETWaitForFixation
        - Prime
        - Stimulus
        - Feedback
    - Goodbye
    - TPLStopRecording
    - TPLClose
    - TETClose
  - Full Script
  - Unreferenced E-Objects

Task 10: Add the TPLSetDisplayEvent PackageCall to Send Display Events to Tobii Pro Lab

Add a PackageCall to the TrialProc that sends display events to Tobii Pro Lab.

When running a Tobii Pro Lab-enabled experiment, it is often useful for later data analysis to send over display information of the stimuli presented in E-Prime to Tobii Pro Lab. The PackageCall **TPLSetDisplayEvent** is designed for this purpose. This PackageCall makes the Stimulus Display available to Tobii Pro Lab for analysis and visualization purposes. For example, if an image of a face is presented on a Slide Object, using this PackageCall results in the same image being displayed with the eye tracking data in the Tobii Pro Lab recording. This PackageCall is placed at the beginning of a Procedure to identify the display events at the start of the trial.

Display objects that are not sent to Tobii Pro Lab using this PackageCall are represented in both the playback and in the thumbnail timeline as "**placeholder**" events." For example, three objects may exist in a Trial Procedure: Stimulus, Fixation, and Feedback. If TPLSetDisplayEvent is used to send over Stimulus display information only, then Fixation, Feedback, and the inter-trial interval are represented in the recording as "placeholders." For more information on Placeholders, please view *Article 29473: Setting Placeholders in Tobii Pro Lab-enabled experiments*.

The first parameter in TPLSetDisplayEvent is the current experiment **Context**.

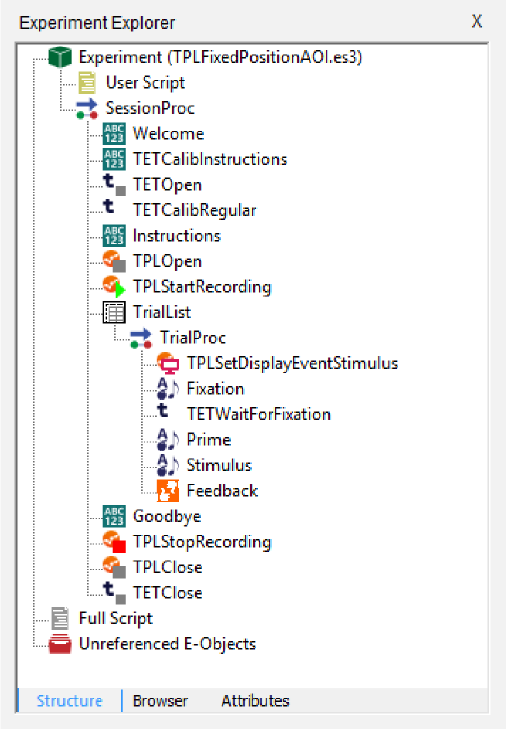
The second parameter (**strMediaName**) is required. This parameter specifies the name of the media file as shown in Tobii Pro Lab. If the Stimulus Display's content is defined by an image or movie file, then we recommend keeping the strMediaName parameter identical to the file name used by the display object. Additionally, we recommend that both the .Filename property as well as this parameter be specified using an Attribute from a List. Alternatively, if using a TextDisplay, SlideButton/Choice/Slider, or a SlideImage/Movie frame to present the stimulus, we recommend the strMediaName to equal the same attribute value from a List that dynamically defines the object's specific stimulus.

The third parameter (**strObject**) is required. The name of the object or sub-object that is to be sent to Tobii Pro Lab needs to be specified with this parameter. The name of the Slide, ImageDisplay, TextDisplay, or MovieDisplay can be entered here (e.g., "Stimulus" would be entered for a Slide named Stimulus). Additionally, Slide sub-objects can be specified (e.g., "Stimulus.Image1" to specify Image1 on the Stimulus Slide).

The fourth parameter (**vMode**) is optional. The allowed values for this parameter are "Fullscreen", "Object", or "File". "Fullscreen" sends a screenshot of the display to Tobii Pro Lab. "Object" sends an image of the Slide sub-object to Tobii Pro Lab. Lastly, "File" sends the entire file (e.g., specified using the .Filename property) that is used as a stimulus to Tobii Pro Lab.

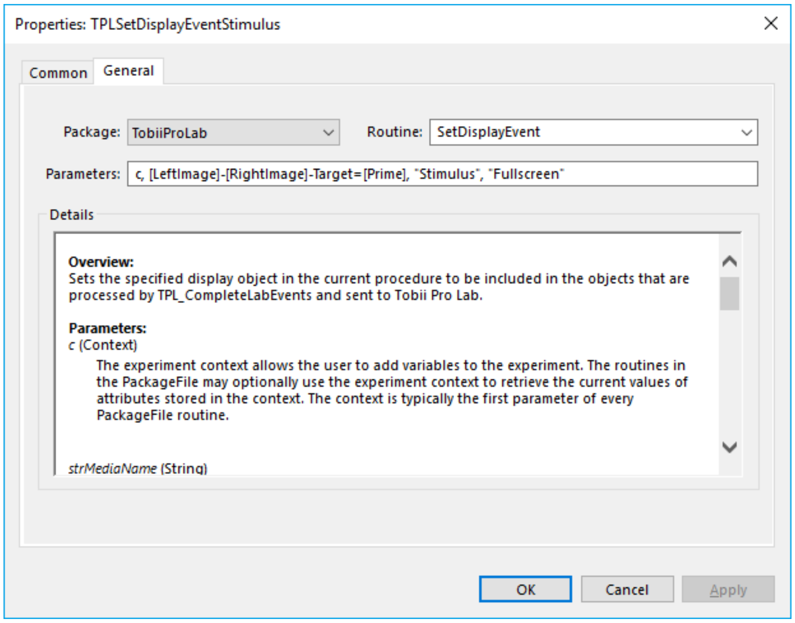
The fifth parameter (**vDelay**) is optional. If specified, it defines the delay in milliseconds after the specified strObject's OnsetTime that elapses before the display contents are captured to a media file. This allows changes in the display object to be reflected in the Tobii Pro Lab recording. To see both the original display as well as the modified one, another call to SetDisplayEvent with vDelay = 0 is required.

1) ***Drag*** the shortcut **TPLSetDisplayEvent** from the **Tobii Pro Toolbox** and ***drop*** it as the **first** object in the **TrialProc.**



2) ***Open*** the object's **Property Pages**.

3) ***Edit*** the **Parameters** field to be **c, [LeftImage]-[RightImage]-Target=[Prime], "Stimulus"**.



Package: TobiiProLab

Routine: SetDisplayEvent

Parameters: c, [LeftImage]-[RightImage]-Target=[Prime], "Stimulus", "Fullscreen"

Details

Overview:
Sets the specified display object in the current procedure to be included in the objects that are processed by TPL_CompleteLabEvents and sent to Tobii Pro Lab.

Parameters:
c (Context)

The experiment context allows the user to add variables to the experiment. The routines in the PackageFile may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every PackageFile routine.

strMediaName (String)

4) ***Click*** the **OK** button to accept the changes.

Task 11: Add the TPLSetAOI PackageCall to Send AOIs to Tobii Pro Lab

Add a PackageCall to the TrialProc that sends AOIs to Tobii Pro Lab.

One features of Tobii Pro Lab is the ability to set Areas of Interests (AOIs) on display events. E-Prime Extensions for Tobii Pro 3.2 allows users to define AOIs during an experiment so they can be used at a later point in Tobii Pro Lab data analysis. The creation of up to 10 AOIs on a single display is supported by the **TPLSetAOI** PackageCall. However, more than 10 AOIs can be created with multiple TPLSetAOI PackageCalls. A rectangular-shaped AOI is created from the frame coordinates of the specified object or sub-object. AOIs can also be defined by either a set of four coordinates that define a rectangle (not based on a frame coordinates) or as a set of coordinates that form a polygon (see the SetAOIR and SetAOIP routines in the *E-Prime Command Reference* for details).

The first parameter is the current experiment Context.

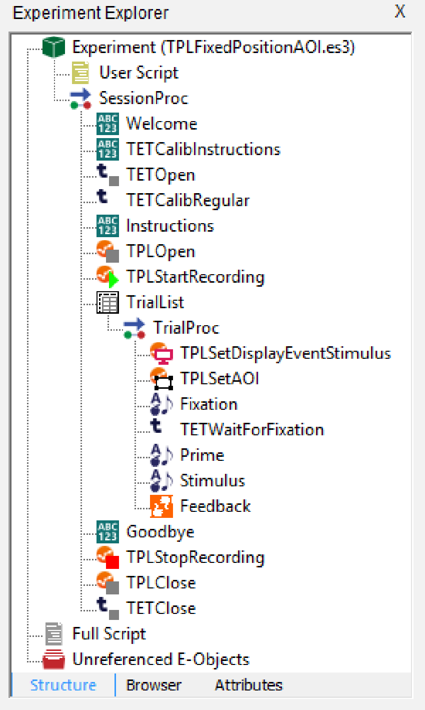
The second parameter (**strMediaName**) is required. This is an identical parameter to the one discussed in *Task 10: Add the TPLSetDisplayEvent PackageCall to Send Display Events to Tobii Pro Lab.*

The third parameter (**strAOIName1**) is required. This parameter is a string value of what the AOI is going to be named in Tobii Pro Lab. For example, if a researcher wanted to specify an eye on an image of a person as an AOI then either “LeftEye” or “RightEye” could be specified.

The fourth parameter (**strObject1**) is required. This is a similar parameter to the one discussed in *Task 10: Add the TPLSetDisplayEvent PackageCall to Send Display Events to Tobii Pro Lab*. The name of the object or sub-object that is defined as an AOI needs to be specified in this parameter. For example, if an AOI needs to be defined on a SlideImage named “Image1”, then “Stimulus.Image1” needs to be specified as the parameter (i.e., the Slide is named Stimulus in this example).

The remaining parameters are all optional. These parameters (**vAOIName2**, **vObject2**, etc.) are necessary if more than one AOI need to be created in Tobii Pro Lab. These parameters work identically to the related parameter mentioned above.

1) ***Drag*** the shortcut **TPLSetAOI** from the **Tobii Pro Toolbox** and ***drop*** it after **TPLSetDisplayEvent** in the **TrialProc.**

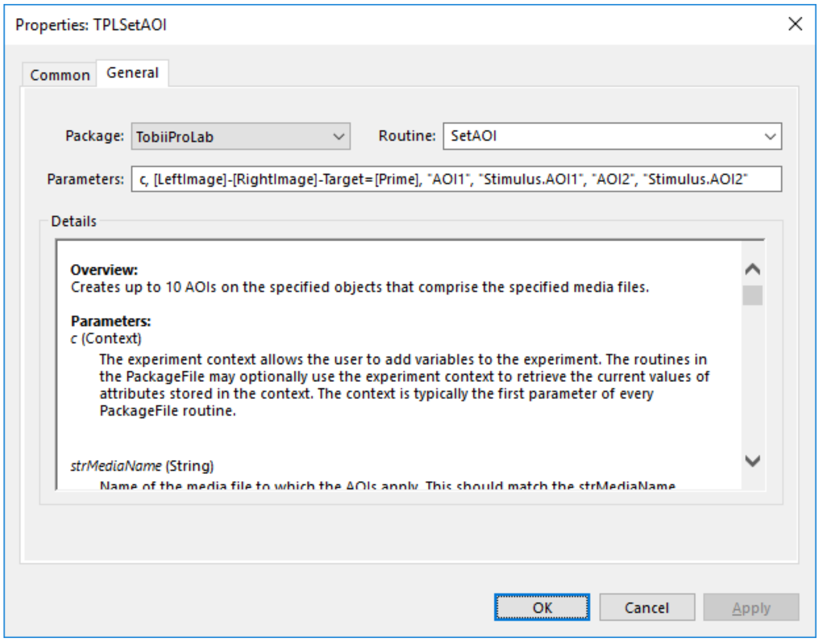


Experiment Explorer

- Experiment (TPLFixedPositionAOI.es3)
  - User Script
  - SessionProc
    - Welcome
    - TETCalibInstructions
    - TETOpen
    - TETCalibRegular
    - Instructions
    - TPLOpen
    - TPLStartRecording
    - TrialList
      - TrialProc
        - TPLSetDisplayEventStimulus
        - TPLSetAOI
        - Fixation
        - TETWaitForFixation
        - Prime
        - Stimulus
        - Feedback
    - Goodbye
    - TPLStopRecording
    - TPLClose
    - TETClose
  - Full Script
  - Unreferenced E-Objects

2) ***Open*** the object’s **Property Pages**.

3) ***Edit*** the **Parameters** field to be **c, [LeftImage]-[RightImage]-Target=[Prime], "AOI1", "Stimulus.AOI1", "AOI2", "Stimulus.AOI2".**



Properties: TPLSetAOI

Package: TobiiProLab

Routine: SetAOI

Parameters: c, [LeftImage]-[RightImage]-Target=[Prime], "AOI1", "Stimulus.AOI1", "AOI2", "Stimulus.AOI2"

Details

Overview:
Creates up to 10 AOIs on the specified objects that comprise the specified media files.

Parameters:
c (Context)

The experiment context allows the user to add variables to the experiment. The routines in the PackageFile may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every PackageFile routine.

strMediaName (String)

Name of the media file to which the AOIs apply. This should match the strMediaName

4) ***Click*** the **OK** button to accept the changes.

Task 12: Add the TPLSetTags PackageCall to Send Tags to Tobii Pro Lab

Add a PackageCall to the TrialProc that sends Tags to Tobii Pro Lab.

When a Tobii Pro Lab-enabled experiment is configured to send AOIs, it is often useful to send Tags to Tobii Pro Lab as well. Tags make it possible to assign conditions to AOIs. Tag creation is possible in E-Prime Extensions for Tobii Pro 3.2 with the **TPLSetTag** PackageCall.

The first parameter is the current experiment Context.

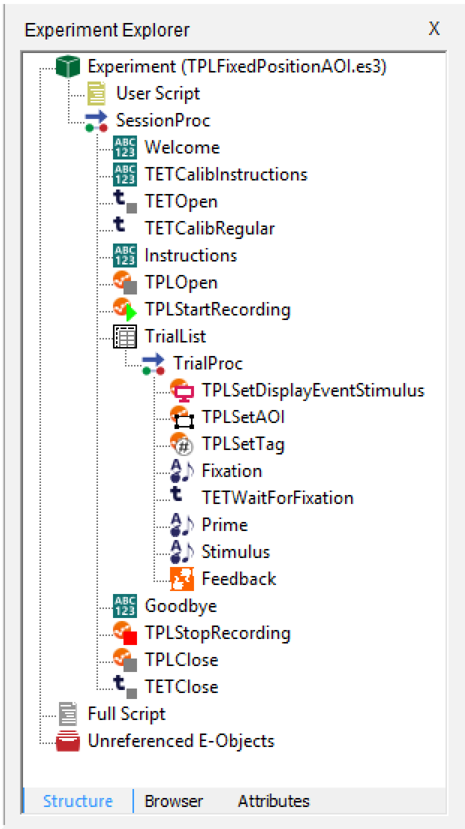
The second parameter (**strMediaName**) is required. This is an identical parameter to the one discussed in *Task 10: Add the TPLSetDisplayEvent PackageCall to Send Display Events to Tobii Pro Lab*.

The third parameter (**strAOIName1**) is required. This is an identical parameter to the one discussed in *Task 11: Add the TPLSetAOI PackageCall to Send AOIs to Tobii Pro Lab*.

The third parameter (**strTagValues1**) is required. This parameter is a string value that identifies the name of the Tag with an optional Group name. Grouped Tags are values assigned within a group. For example, “Eye Color” could be the name of the group with the Tag values being “Blue”, “Green”, “Brown”, etc. In this parameter, "EyeColor.Blue” would need specified for the “Blue” tag. “EyeColor” is the name of the Group, and “Blue” is one of the values. To enter a tag that is not a part of a group (i.e., Ungrouped Tags), the string name of the tag needs to be entered. Using the example above, if a researcher did not want to create a Group Tag for eye color, “Blue”, “Green”, and “Brown” could be entered without the “EyeColor.” prefix.

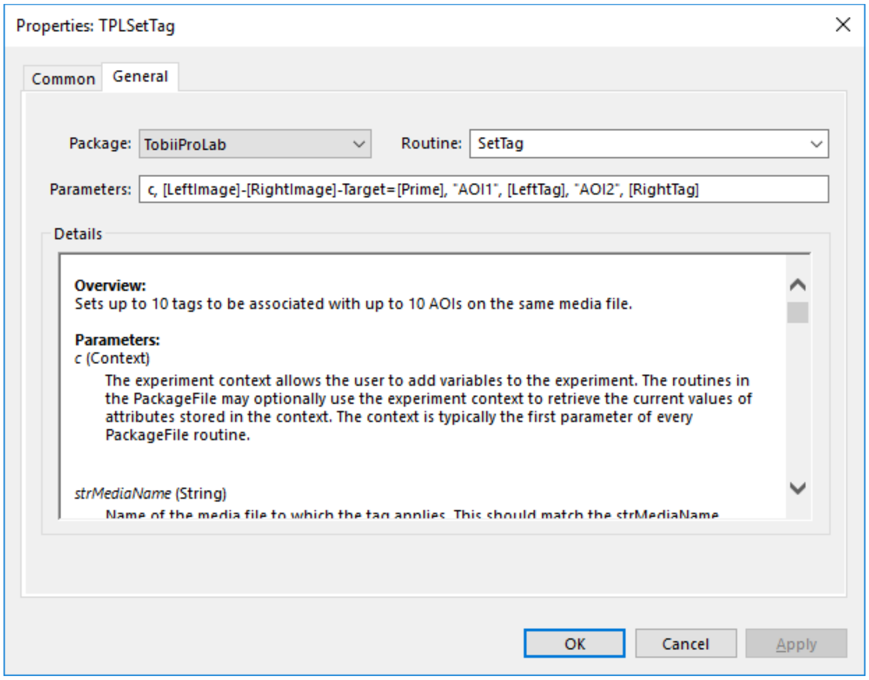
The rest of the parameters (**vAOIName2**, **vTagValues2**, etc**.**) are all optional. These parameters are necessary if more than one Tag need to be created in Tobii Pro Lab. These parameters work identically to the related parameter mentioned above. 10 Tags can be created with the TPLSetTag PackageCall, however, more can be created with multiple TPLSetTag PackageCalls.

1) ***Drag*** the shortcut **TPLSetTag** from the **Tobii Pro Toolbox** and ***drop*** it after **TPLSetAOI** in the **TrialProc.**



2) ***Open*** the object’s **Property Pages**.

3) ***Edit*** the **Parameters** field to be **c, [LeftImage]-[RightImage]-Target=[Prime], "AOI1", [LeftTag], "AOI2", [RightTag]**.



Package: TobiiProLab

Routine: SetTag

Parameters: c, [LeftImage]-[RightImage]-Target=[Prime], "AOI1", [LeftTag], "AOI2", [RightTag]

Details

Overview:
Sets up to 10 tags to be associated with up to 10 AOIs on the same media file.

Parameters:
c (Context)

The experiment context allows the user to add variables to the experiment. The routines in the PackageFile may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every PackageFile routine.

strMediaName (String)

Name of the media file to which the tag applies. This should match the strMediaName

NOTE: *[LeftTag] and [RightTag] are Attributes defined in the TrialList for sending Tags to Tobii Pro Lab. The values for the Attributes are either “Target” or “Non-Target.”*

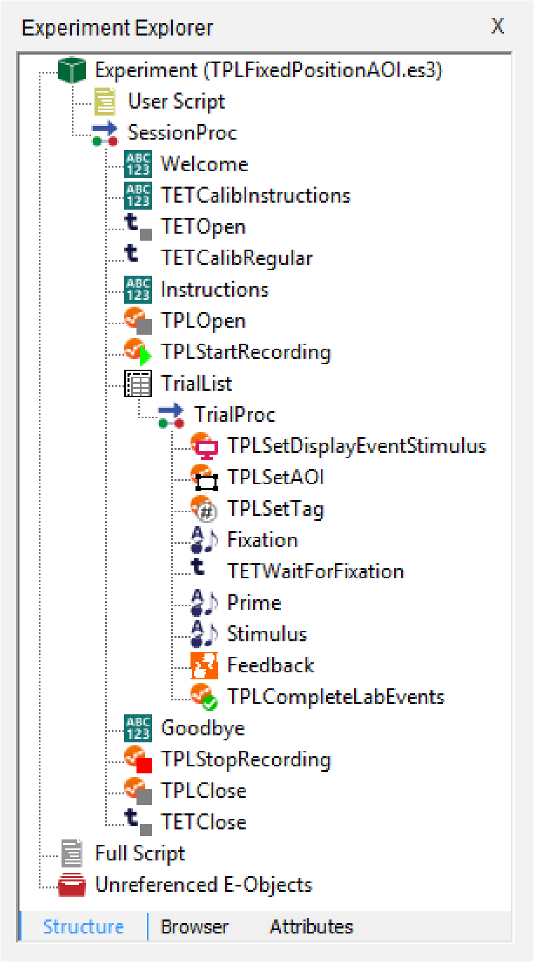
4) ***Click*** the **OK** button to accept the changes.

Task 13: Add the TPLCompleteLabEvents PackageCall to Finish the Display Event

Add a PackageCall to the TrialProc that completes sending the display events to Tobii Pro Lab.

At the end of a trial it is necessary to utilize the **TPLCompleteLabEvents** PackageCall. This PackageCall signifies the completion of the trial by sending the Stimulus Display information, AOIs, and Tags to Tobii Pro Lab. This PackageCall should be placed at a non-time critical portion of the Procedure it is in (e.g., the end of the Procedure).

1) ***Drag*** the shortcut **TPLCompleteLabEvents** from the **Tobii Pro Toolbox** and ***drop*** it as the last object in the **TrialProc**.



```
Experiment (TPLFixedPositionAOI.es3)
    User Script
    SessionProc
        Welcome
        TETCalibInstructions
        TETOpen
        TETCalibRegular
        Instructions
        TPLOpen
        TPLStartRecording
        TrialList
            TrialProc
                TPLSetDisplayEventStimulus
                TPLSetAOI
                TPLSetTag
                Fixation
                TETWaitForFixation
                Prime
                Stimulus
                Feedback
                TPLCompleteLabEvents
        Goodbye
        TPLStopRecording
        TPLClose
        TETClose
Full Script
Unreferenced E-Objects
```

Task 14: Prepare the Tobii Pro Lab Project

Open Tobii Pro Lab and prepare a Project.

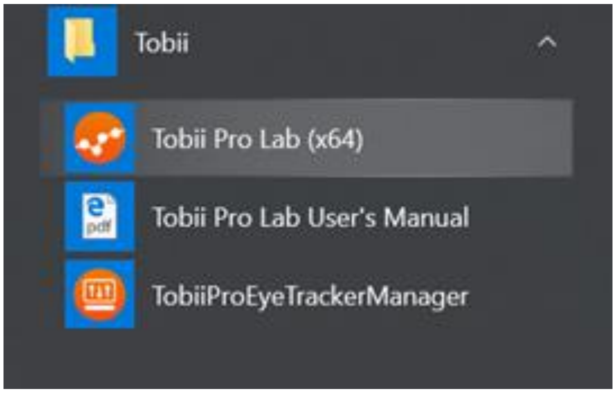
Before a Tobii Pro Lab-enabled experiment can be run, the Tobii Pro Lab software application must be opened. EET 3.2 requires that the Tobii Pro Lab projects are External Presenter Project types. Once a Project has been selected, named, and created the Project Overview screen appears. Researchers must select the “Record” tab in Tobii Pro Lab before running the experiment from E-Studio.

Tobii Pro Lab and the Tobii Pro Lab-enabled experiments can be run on either one or two machines (please refer to *System Requirements*). We recommend preparing Tobii Pro Lab prior to a test session when running on either configuration. For more information on these configurations please view *Appendix B: Hardware Configurations*.

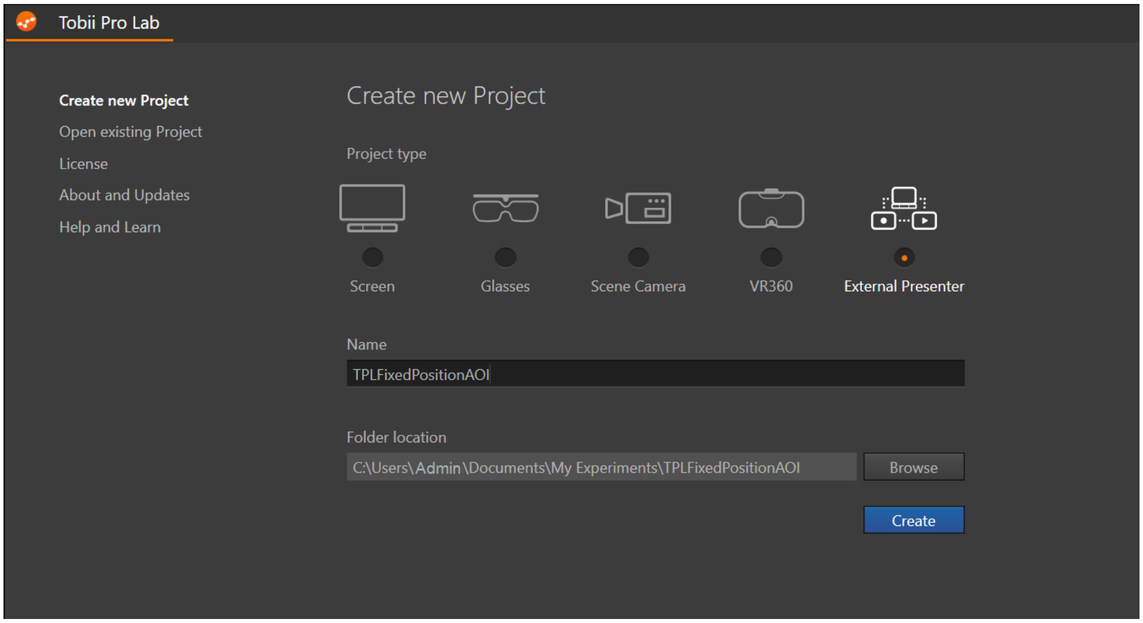
NOTE: *Users should not interact with Tobii Pro Lab on a two-machine setup after a test session has begun. This can result in early experiment termination and/or potential loss of data.*

NOTE: *Please refer to Tobii Pro Resources for the link to the Tobii Pro Lab website which provides downloads and information on installation and use.*

1) ***Click*** on the **Windows Start** menu, ***search*** for **Tobii**. From the menu, ***click*** on **Tobii Pro Lab** to launch the application.



2) ***Select*** the **External Presenter** Project type. ***Specify*** "**TPLFixedPositionAOI**" as the **Project Name**. ***Click*** the **Create** button to ***create*** the **Project**.



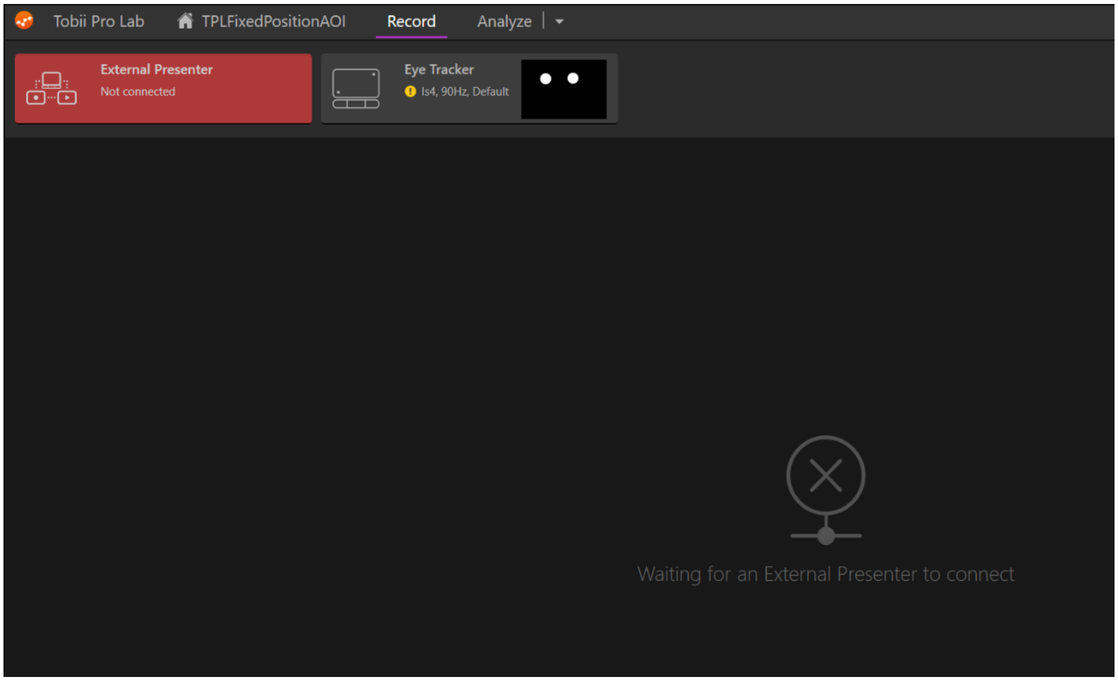
Create new Project

Project type: External Presenter

Name: TPLFixedPositionAOI

Folder location: C:\Users\Admin\Documents\My Experiments\TPLFixedPositionAOI

3) ***Select*** the **Record** tab to finalize the Tobii Pro Lab Project preparation.



External Presenter
Not connected

Eye Tracker
Is4, 90Hz, Default

Waiting for an External Presenter to connect

NOTE: *The eye tracker shown on the Record tab should match the eye tracker specified in the corresponding E-Prime experiment.*

Task 15: Run the Experiment

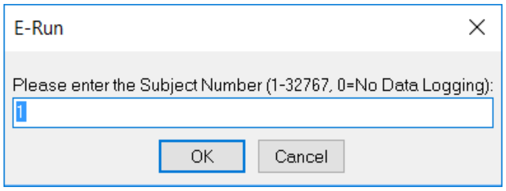
Run the experiment to verify that the eye tracker is working.

The basic steps necessary to create a Tobii Pro Lab-enabled paradigm are now completed. Tobii Pro Lab-enabled experiments can be run locally from E-Studio during development and testing if Tobii Pro Lab is configured for the Project. Experiments should always be tested prior to scheduling participants and data collection.

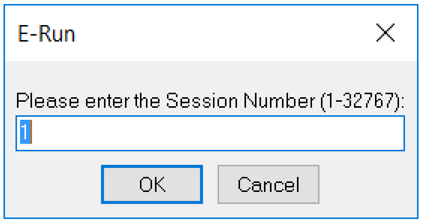
Running the Tobii Pro Lab-enabled experiment should occur to verify the display information, AOIs, and tags exist in Tobii Pro Lab after the test session.

1) ***Click*** the **Run icon** on the toolbar to generate and run the experiment locally.

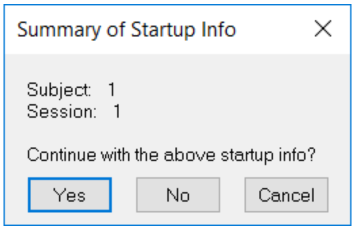
2) ***Press*** **Enter** to ***accept*** the **default values** for each of the initialization prompts presented.



Please enter the Subject Number (1-32767, 0=No Data Logging):
1



Please enter the Session Number (1-32767):
1



Subject: 1
Session: 1

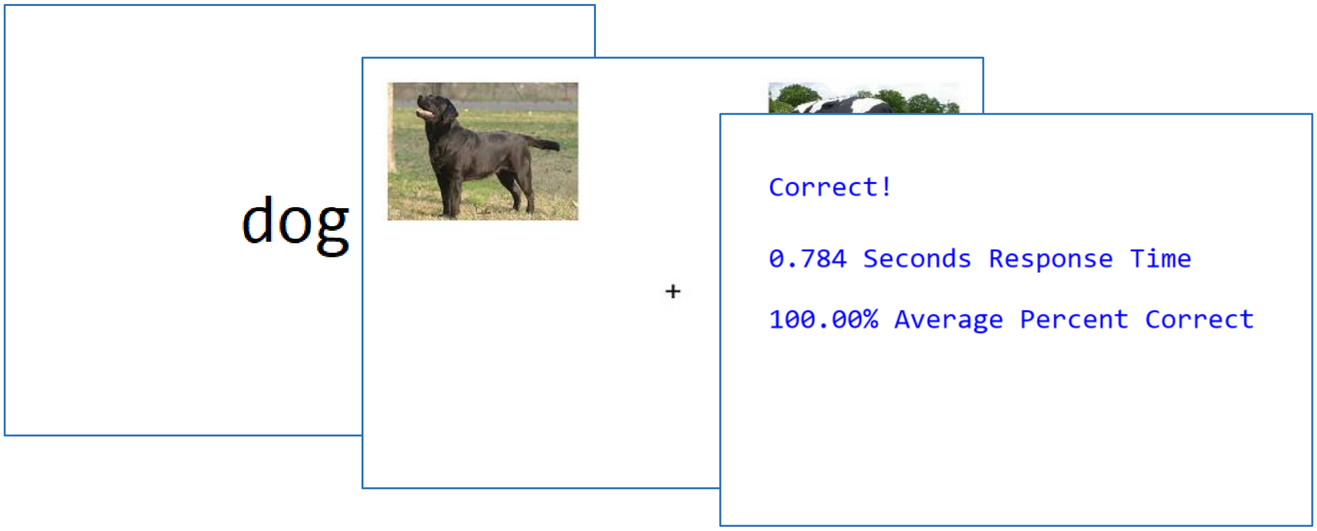
Continue with the above startup info?

3) ***Follow*** the **prompts on screen** to ***complete*** the **calibration** sequence.

4) ***Accept*** the **calibration (if acceptable)** and ***perform*** the **experiment**.

NOTE: *Refer to Tutorial 1, Task 15: Run the Experiment for information on good/bad calibration.*

5) ***Observe*** the **stimulus presentation sequence** to verify the experiment is functioning correctly and completes with no errors being generated.



```
Correct!

0.784 Seconds Response Time

100.00% Average Percent Correct
```

Task 16: Verify your Display Stimuli were sent to Tobii Pro Lab

Verify that the recording, display stimuli, AOIs, and Tags are in Tobii Pro Lab.

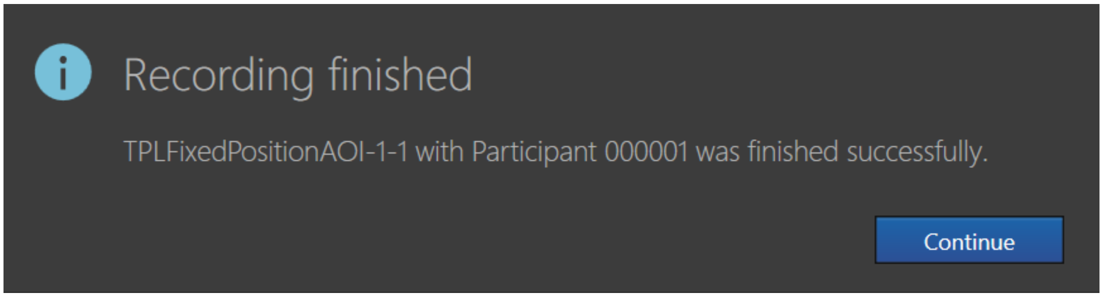
After running a Tobii Pro Lab-enabled experiment, we strongly recommend verifying the data was sent to Tobii Pro Lab. If the recording of the test session was successful, a notification exists in Tobii Pro Lab stating that the recording is finished. After dismissing the notification, the recording can be accessed for analysis and visualization purposes.

Next to the Record tab in Tobii Pro Lab, the Analyze tab exists. Clicking on the dropdown arrow next to Analyze shows all available recordings in the Project and various analysis features. The goal of this task is to briefly show what information E-Prime sent to Tobii Pro Lab. For more in-depth details of performing an analysis in Tobii Pro Lab, please refer to the Tobii Documentation.

NOTE: *Tobii Pro Lab does not "record" the screen of the experiment, but rather attempts to recreate a representation of the stimulus display based on the stimulus/event related information (i.e., with TPLSetDisplayEvents) that is sent to it by E-Prime. In some cases, this recreation may not be identical to the display observed by the participant but is the best representation of the display events.*

1) ***Navigate*** to **Tobii Pro Lab** after running a Tobii Pro Lab-enabled experiment.

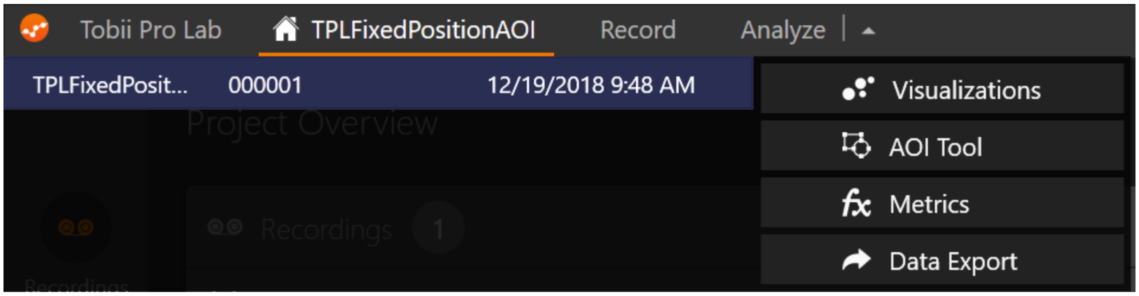
2) ***Verify*** a **notification** appears stating that **the recording has finished successfully**. ***Click*** the **Continue** button.



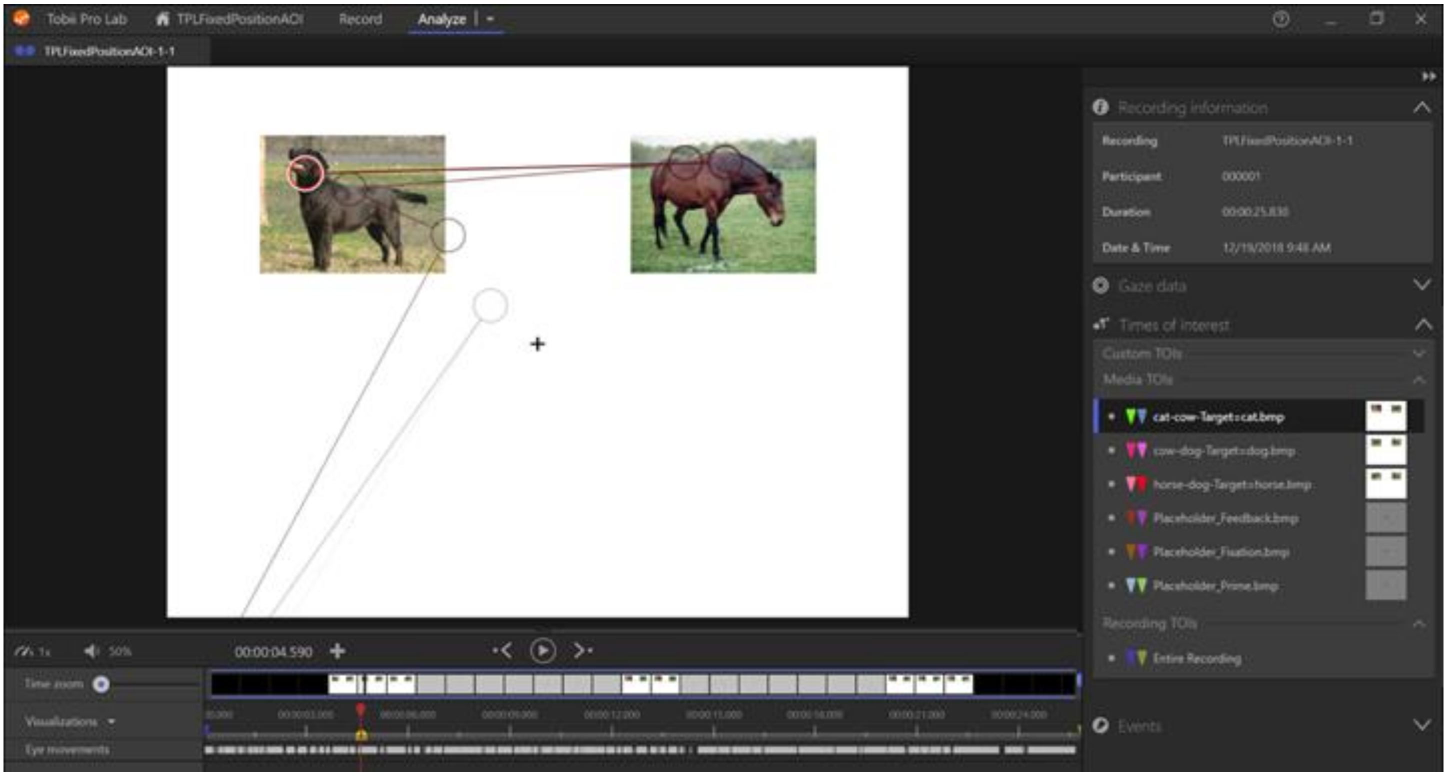
Recording finished

TPLFixedPositionAOI-1-1 with Participant 000001 was finished successfully.

3) ***Click*** the **Analyze dropdown** arrow. ***Double click*** on the **recording.**



4) ***Click*** the **Play** button to watch a **playback of the recording**. ***Verify*** the **Stimulus Display events** were sent to Tobii Pro Lab. These can be viewed under Times of Interest (e.g., cat-cow-Target=cat.bmp).

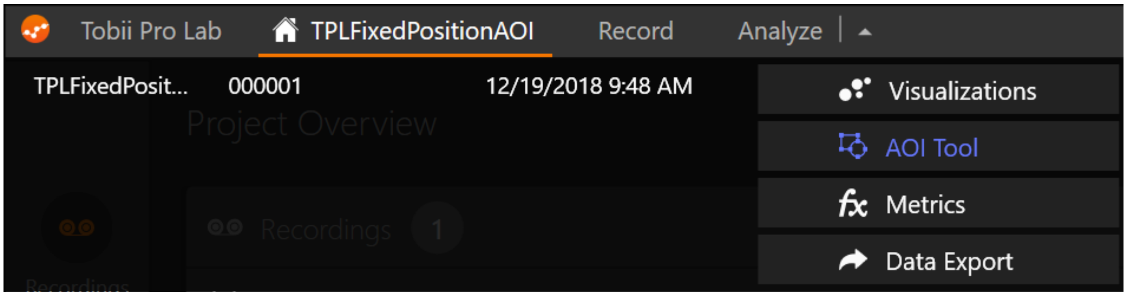


Recording: TPLFixedPositionAOI-1-1
Participant: 000001
Duration: 00:00:25.830
Date & Time: 12/19/2018 9:48 AM

- cat-cow-Target=cat.bmp
- cow-dog-Target=dog.bmp
- horse-dog-Target=horse.bmp
- Placeholder_Feedback.bmp
- Placeholder_Fixation.bmp
- Placeholder_Prime.bmp
- Entire Recording

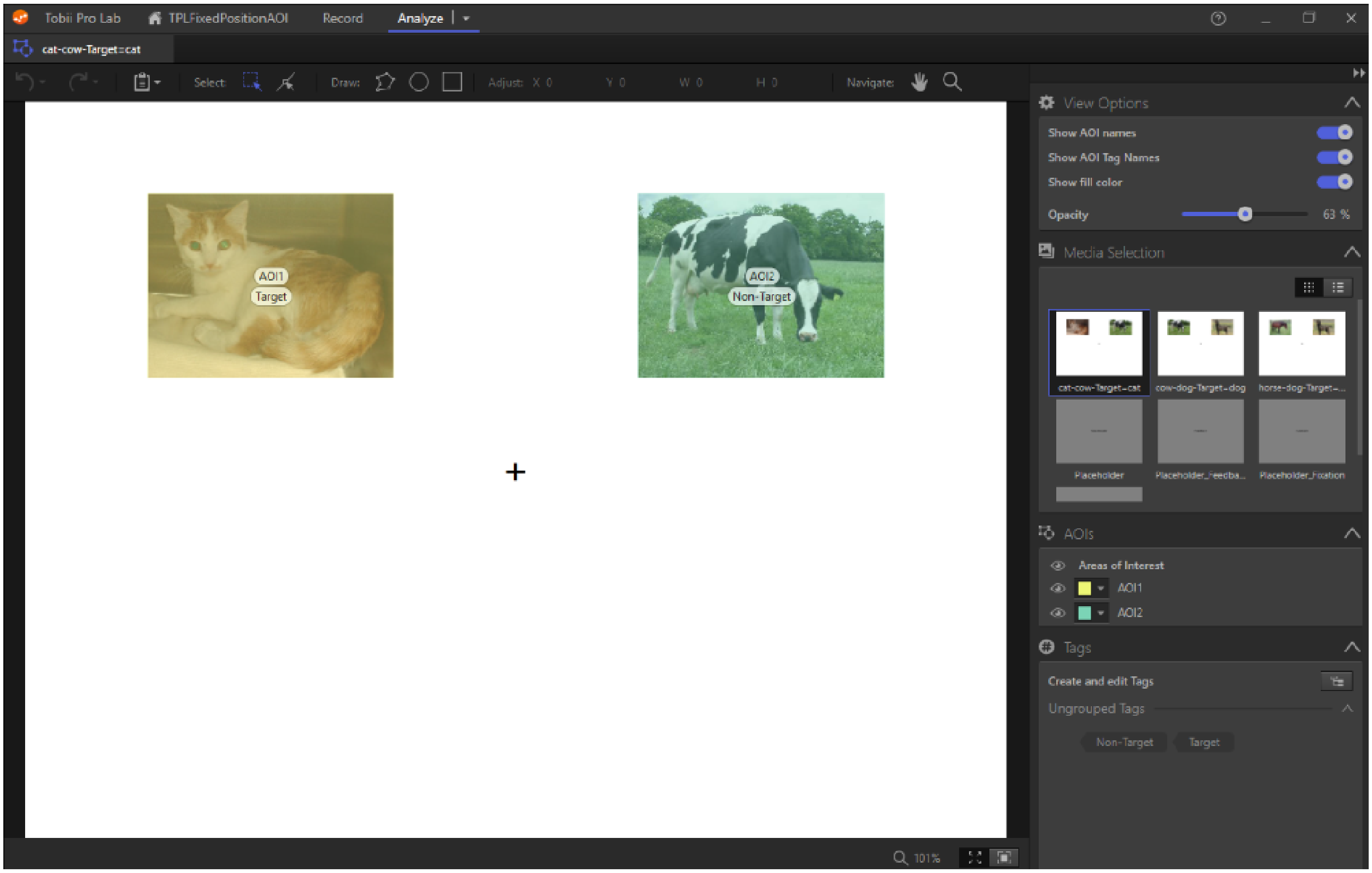
NOTE: *Additionally, "Placeholder" images exist under Media TOIs. These images represent the time in the experiment where the stimulus information was not specifically recorded.*

5) ***Click*** the **Analyze dropdown** arrow. ***Click*** on the **AOI Tool** option**.**



TPLFixedPosit... 000001 12/19/2018 9:48 AM

6) ***Verify*** both the **AOIs** and **Tags** were sent to Tobii Pro Lab.



Show AOI names: on

Show AOI Tag Names: on

Show fill color: on

Opacity: 63 %

Areas of Interest

- AOI1
- AOI2

Ungrouped Tags: Non-Target, Target

4. Troubleshooting

The following topics are common issues and solutions to help you when implementing and testing experiments with Tobii Pro. In the case that one of the topics below cannot fix your issue, Psychology Software Tools recommends saving and closing the experiment. If any issue persists please contact support (See Chapter 1, Section 1.3: *Product Service and Support*).

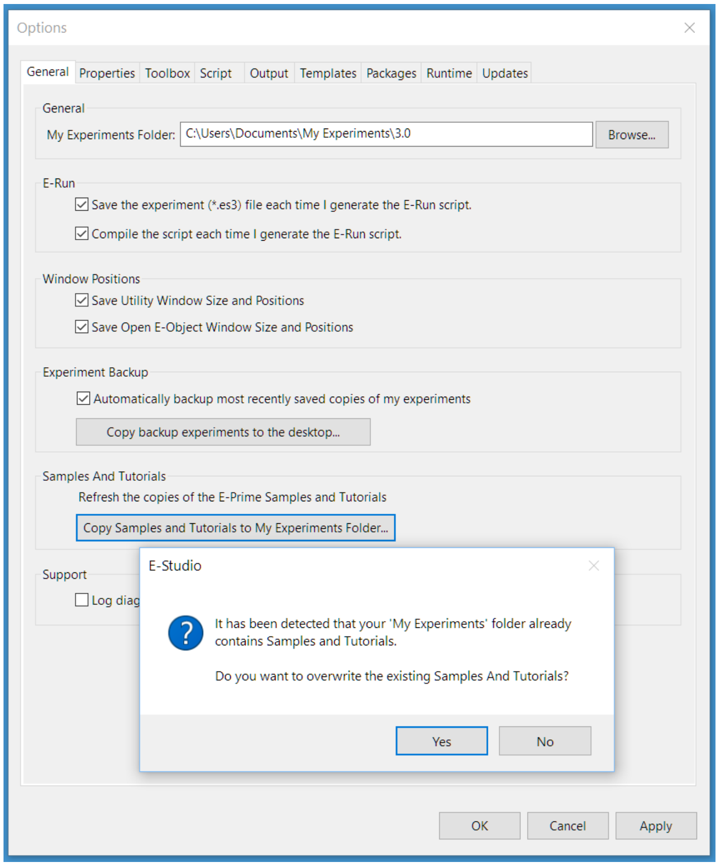
4.1. Cannot locate E-Prime Samples and Tutorials

When you launch E-Studio for the first time after installing E-Prime Extensions for Tobii Pro 3.2, you are prompted to update the Samples and Tutorials experiments. If you choose to not update now, you can force another update of the Samples and Tutorials at any time from within E-Studio. Forcing a refresh of the experiments can be helpful if you cannot locate the Samples and Tutorials. This feature is also helpful if you make changes to a sample or tutorial but then wish to revert to the original sample or tutorial.

1) ***Open*** **E-Studio** and ***select*** **Tools** > **Options**… from the application menu bar.

2) ***Click*** on the "**Copy Samples and Tutorials to …My Experiments Folder…**" button at the bottom of the Options dialog.

3) ***Click*** on **Yes** if prompted to overwrite the Samples and Tutorials.



General

My Experiments Folder: C:\Users\Documents\My Experiments\3.0

E-Run

☑ Save the experiment (*.es3) file each time I generate the E-Run script.

☑ Compile the script each time I generate the E-Run script.

Window Positions

☑ Save Utility Window Size and Positions

☑ Save Open E-Object Window Size and Positions

Experiment Backup

☑ Automatically backup most recently saved copies of my experiments

Samples And Tutorials

Refresh the copies of the E-Prime Samples and Tutorials

Support

□ Log diag

E-Studio

It has been detected that your 'My Experiments' folder already contains Samples and Tutorials.

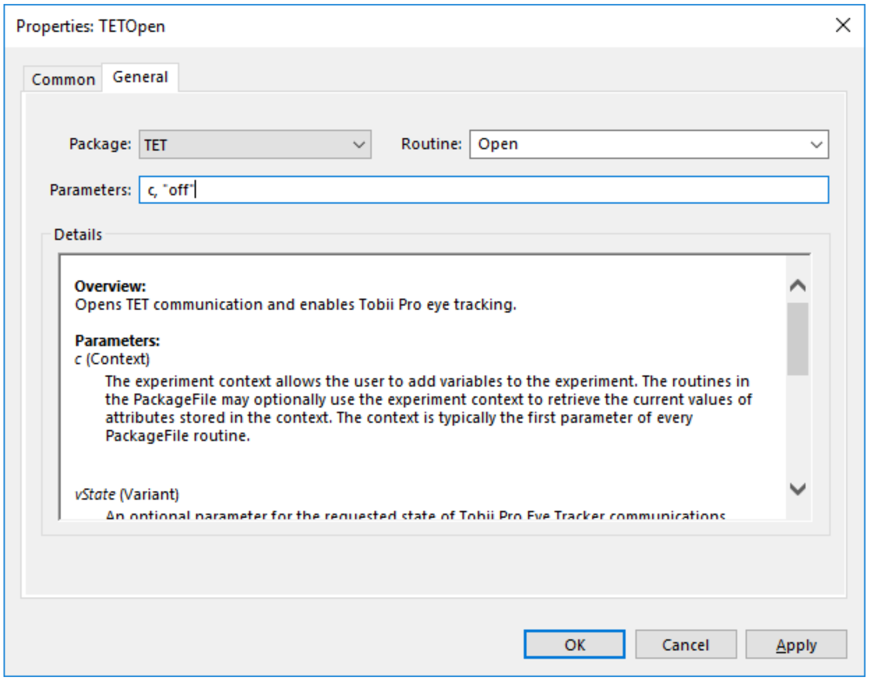
Do you want to overwrite the existing Samples And Tutorials?

4.2. Running without an eye tracker

We recommend testing Tobii Pro-enabled experiments internally before testing with participants. Running without an eye tracker attached is an effective way to test an experiment without all necessary equipment. By default, an error occurs when running a Tobii Pro-enabled experiment without an eye tracker attached. To correct the error, the TETOpen PackageCall needs to be edited.

1) ***Double click*** the PackageCall **TETOpen** to open its properties.

2) ***Edit*** the **Parameters** field to be **c, "off".**



Package: TET

Routine: Open

Parameters: c, "off"

Overview:
Opens TET communication and enables Tobii Pro eye tracking.

Parameters:
c (Context)
The experiment context allows the user to add variables to the experiment. The routines in the PackageFile may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every PackageFile routine.

vState (Variant)
An optional parameter for the requested state of Tobii Pro Eye Tracker communications

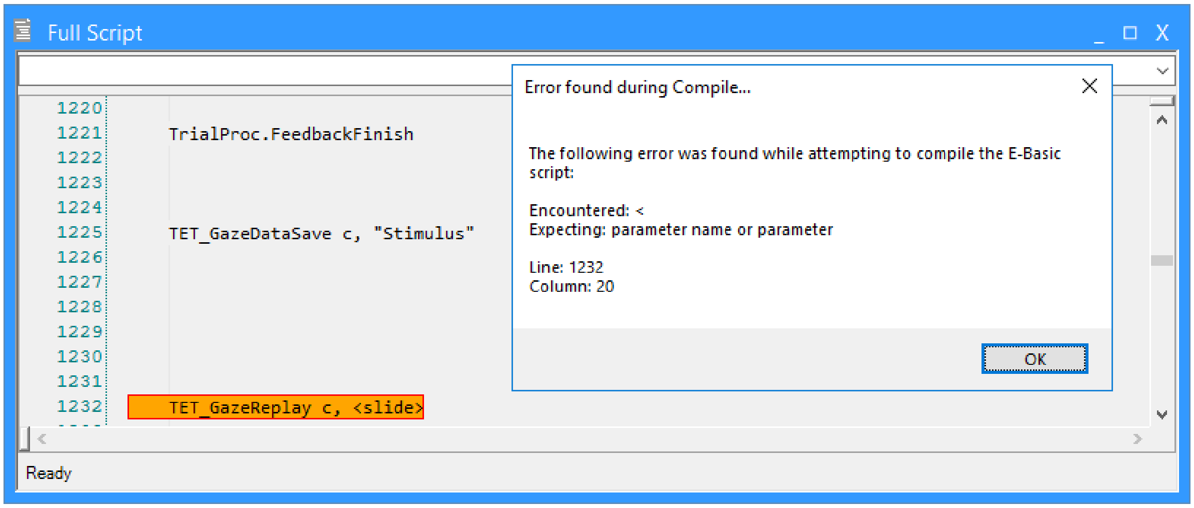
3) ***Click*** the **OK** button to accept the changes.

NOTE: *Additionally, the same steps can be run above for **TPLOpen** in the event an experiment needs to be tested without Tobii Pro Lab.*

4.3. Missing and Incomplete Parameter Errors

Some PackageCalls require additional information specified in the parameters. For example, TETGazeReplay requires one additional parameter to indicate the name of the Slide object.

When designing an experiment, be sure to check the Properties of the PackageCall for any additional parameters. More information is needed when a PackageCall has “<>” for the parameters. The angle brackets (“<>”) are placeholders for more information required about the experiment configuration. The example below shows an error that occurs because the angle brackets were not changed to a correct parameter. The error states that a left angle bracket was encounter. Additionally, the highlighted line in the Full Script window shows that TET_GazeReplay has a parameter (i.e., theSlide) that still has the angle brackets and need changed appropriately.



```
1220
1221     TrialProc.FeedbackFinish
1222
1223
1224
1225     TET_GazeDataSave c, "Stimulus"
1226
1227
1228
1229
1230
1231
1232     TET_GazeReplay c, <slide>
```

Error found during Compile...

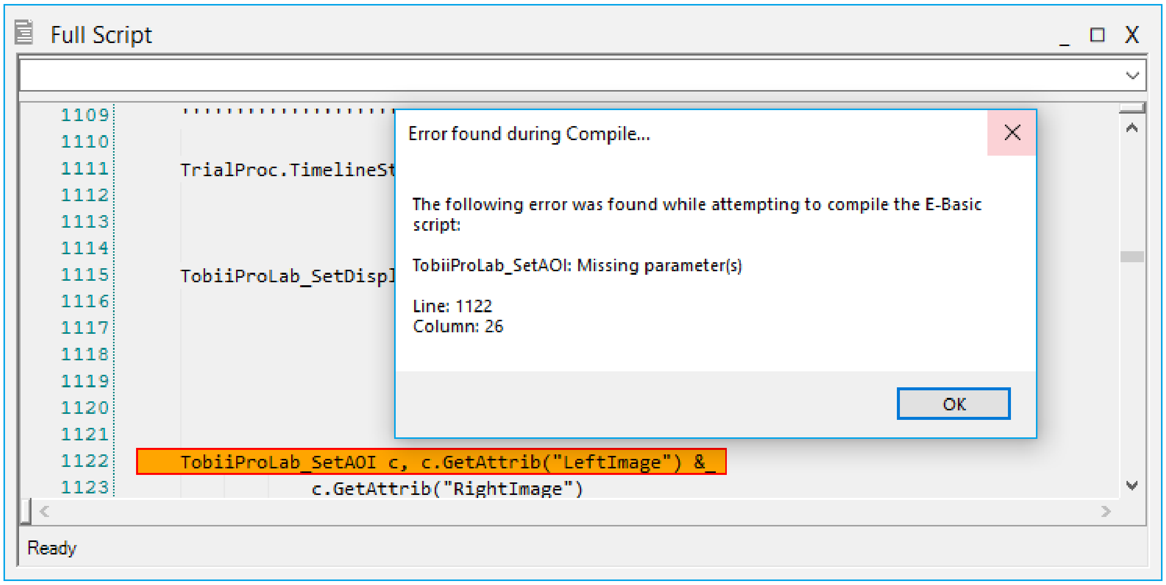
The following error was found while attempting to compile the E-Basic script:

Encountered: <
Expecting: parameter name or parameter

Line: 1232
Column: 20

NOTE: *When designing your experiment, we recommend reading through the Details of the PackageCall’s properties.* ***Details*** *provides descriptions and remarks of all required and optional parameters.*

An error occurs if a required parameter is missing from a Routine. Depending on what is specified in a PackageCall's parameters, the error may either be specific or generic. The example below shows a specific error message that occur. The error states that the PackageCall TPLSetAOI is missing parameters (i.e., strAOIName1 and strObjectName1 are missing). Additionally, the Full Script window highlights the PackageCall for where the error occurs.



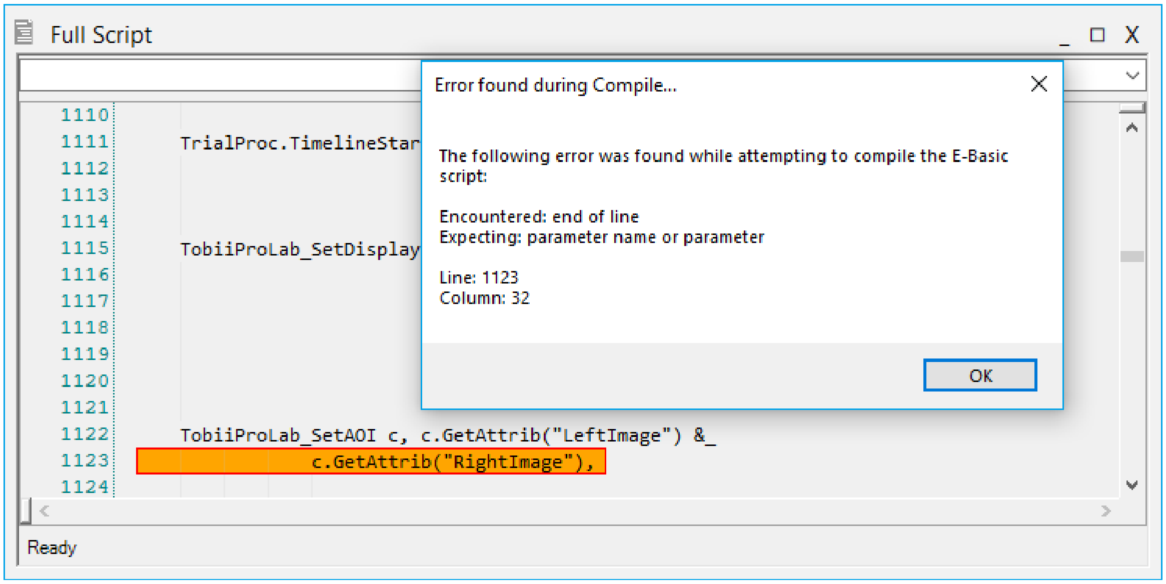
```
    '''''''''''''''''''''
    
    TrialProc.TimelineSt
    
    
    
    TobiiProLab_SetDispl
    
    
    
    
    
    
    TobiiProLab_SetAOI c, c.GetAttrib("LeftImage") &_
                c.GetAttrib("RightImage")
```

The following error was found while attempting to compile the E-Basic script:

TobiiProLab_SetAOI: Missing parameter(s)

Line: 1122
Column: 26

A generic error states that the more specified parameters are expected. The example below shows the error message. Additionally, the Full Script window appears with a line of code highlighted. The highlighted line of code shows an additional comma after ("RightImage"). Due to the comma being placed in the parameters, the error generated is less specific and does not state the PackageCall with the missing parameter. However, the previous line in the Full Script shows that the error happened in the Routine TobiiProLab_SetAOI (i.e., TPLSetAOI).



```
    
    TrialProc.TimelineStar
    
    
    
    TobiiProLab_SetDisplay
    
    
    
    
    
    
    TobiiProLab_SetAOI c, c.GetAttrib("LeftImage") &_
                c.GetAttrib("RightImage"),
    
```

The following error was found while attempting to compile the E-Basic script:

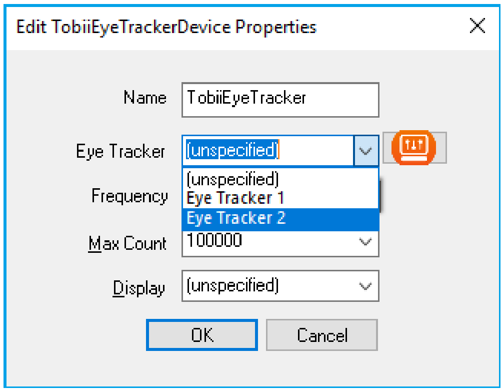
Encountered: end of line
Expecting: parameter name or parameter

Line: 1123
Column: 32

4.4. Unable to Open/Connect to the Eye Tracker

Some additional settings may need to be specified if an error occurs that states “unable to open or connect to the Tobii Eye Tracker” during an experiment.

In E-Prime, be sure that the PackageCall TETOpen has its state configured to “on.” A state of “off” disables communications to the eye tracker. Additionally, ensure that the TobiiEyeTracker Device Properties are configured correctly. For example, if your lab is using multiple eye trackers then the Eye Tracker property should be configured in the TobiiEyeTracker Device Properties.



Edit TobiiEyeTrackerDevice Properties

Name: TobiiEyeTracker

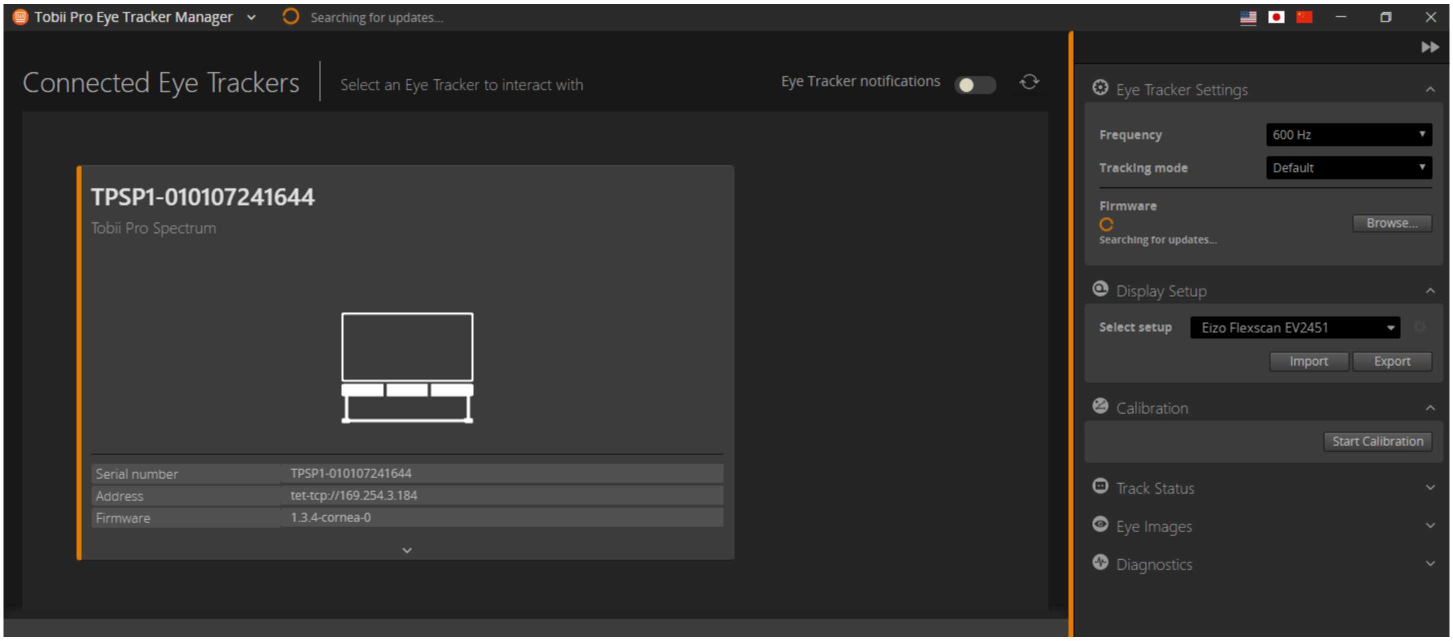
Eye Tracker: (unspecified)

Frequency:

Max Count: 100000

Display: (unspecified)

If the settings in E-Prime are configured successfully and the error still occurs, we recommend ensuring that the eye tracker is appropriately configured and functioning in the Tobii Pro Eye Tracker Manager application. Please refer to the Tobii Pro documentation for information on how to use Tobii Pro Eye Tracker Manager.



Tobii Pro Eye Tracker Manager

Searching for updates...

Connected Eye Trackers | Select an Eye Tracker to interact with

Eye Tracker notifications

TPSP1-010107241644

Tobii Pro Spectrum

Serial number	TPSP1-010107241644
Address	tet-tcp://169.254.3.184
Firmware	1.3.4-cornea-0

Eye Tracker Settings

Frequency: 600 Hz

Tracking mode: Default

Firmware

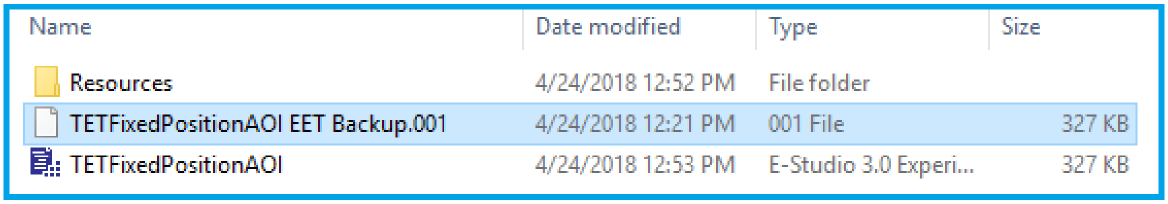
Searching for updates...

Display Setup

Select setup: Eizo Flexscan EV2451

4.5. Upgrading Experiments to E-Prime Extensions for Tobii

Upgrading experiments from EET 2.x, or EET 3.0 is possible. However, it should be noted that once an experiment is brought forward from EET 2.x or EET 3.0 to 3.2, it cannot go back to the previous version. However, EET 3.2 makes a backup of any EET 3.0 file before overwriting the original contents of the EET 3.0 file. The backup file resides in the same location as the original experiment and has "Backup" in the experiment name.



Name	Date modified	Type	Size
Resources	4/24/2018 12:52 PM	File folder	
TETFixedPositionAOI EET Backup.001	4/24/2018 12:21 PM	001 File	327 KB
TETFixedPositionAOI	4/24/2018 12:53 PM	E-Studio 3.0 Experi...	327 KB

NOTE: *No backup file is created for EET 2.x; only EET 3.0.*

Additionally, whenever an experiment is upgraded, saving the gaze data can be updated to use the EET 3.2 functionality (i.e., *Tutorial 2: Logging Gaze Data in an Experiment*). We also recommend verifying the Tobii Eye Tracker properties are configured correctly after upgrading.

4.6. Locate Your IP Address and Ping Another Computer

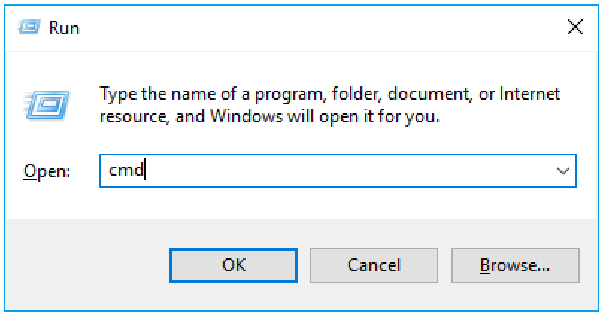
Depending on the paradigm using E-Prime Extensions for Tobii Pro 3.2, it may be necessary to use a multiple machine configuration. When using multiple machines, both machines need to be connected to one another. The simplest way to determine that one machine is connected to another machine is to ping one machine from the other machine. The IP Address of the machine that is being communicated with needs to be known before it can be pinged.

NOTE: *The following steps use Machine 1 and Machine 2 to refer to the machines that need to be connected.*

Perform the following to find the IP address of Machine 2:

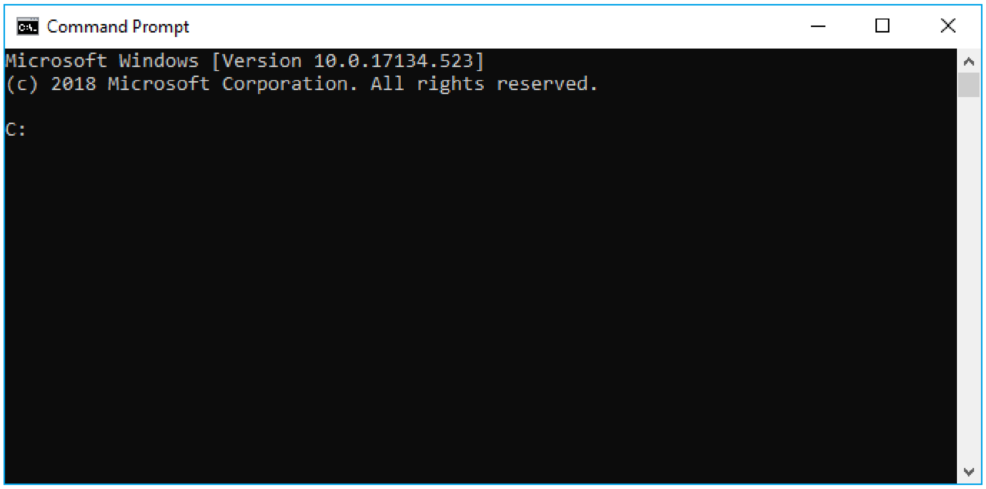
1) ***Hold down*** the **Windows key** and ***press*** the **R key** to open the **Run dialog**.

2)***Type*** "**cmd**" and ***click*** **OK** in the **Run** dialog.


Run
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
Open:
OK
Cancel
Browse...

Open: cmd

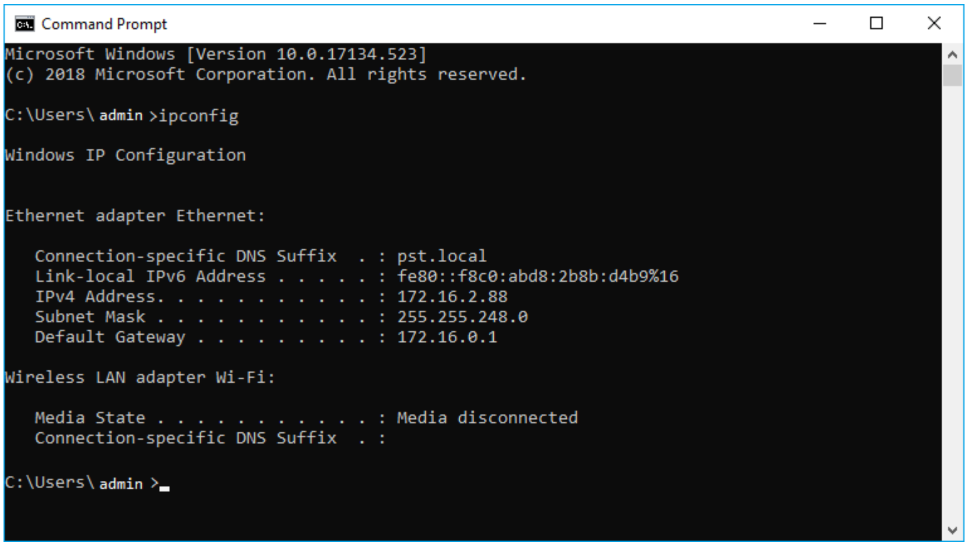
3) ***Verify*** the **Command Prompt** opens.



```
Microsoft Windows [Version 10.0.17134.523]
(c) 2018 Microsoft Corporation. All rights reserved.

C:
```

4) ***Type*** "**ipconfig**" in Command Prompt and ***press*** **Enter**.

5) ***View*** the **IP Address** in the **Command Prompt window**.



```
Microsoft Windows [Version 10.0.17134.523]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ admin >ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : pst.local
   Link-local IPv6 Address . . . . . : fe80::f8c0:abd8:2b8b:d4b9%16
   IPv4 Address. . . . . . . . . . . : 172.16.2.88
   Subnet Mask . . . . . . . . . . . : 255.255.248.0
   Default Gateway . . . . . . . . . : 172.16.0.1

Wireless LAN adapter Wi-Fi:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\ admin >_
```

NOTE: *Not all IP Addresses look the same. In this example, the IP address is IPv4 Address.*

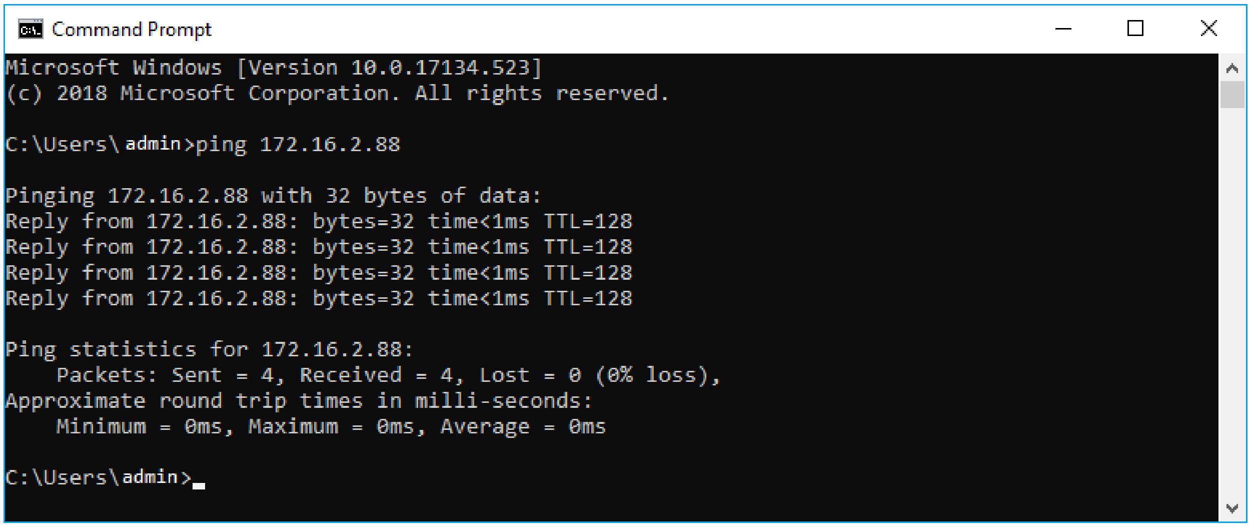
Perform the following on Machine 1 to communicate with Machine 2:

6) ***Hold down*** the **Windows key** and ***press*** the **R key** to open the **Run dialog**.

7)***Type*** "**cmd**" and ***click*** **OK** in the **Run** dialog.

8) ***Verify*** the **Command Prompt** opens.

9) ***Type "*****Ping [the IP Address from Step 5]"** and ***press*** **Enter**.



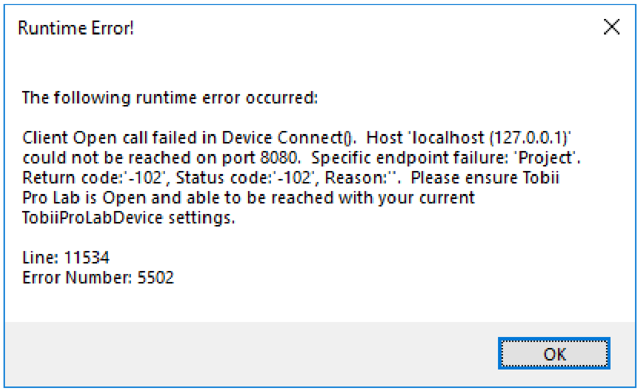
```
Microsoft Windows [Version 10.0.17134.523]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\ admin>ping 172.16.2.88

Pinging 172.16.2.88 with 32 bytes of data:
Reply from 172.16.2.88: bytes=32 time<1ms TTL=128
Reply from 172.16.2.88: bytes=32 time<1ms TTL=128
Reply from 172.16.2.88: bytes=32 time<1ms TTL=128
Reply from 172.16.2.88: bytes=32 time<1ms TTL=128

Ping statistics for 172.16.2.88:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\admin>_
```

If the machines can communicate, a reply message occurs. If the machines cannot communicate, an error message occurs.

4.7. Issues with Improperly Configuring Tobii Pro Lab

Project Type Error

E-Prime Extensions for Tobii Pro 3.2 requires the External Presenter Project Type in Tobii Pro Lab. If another Project Type is selected, the following error occurs:

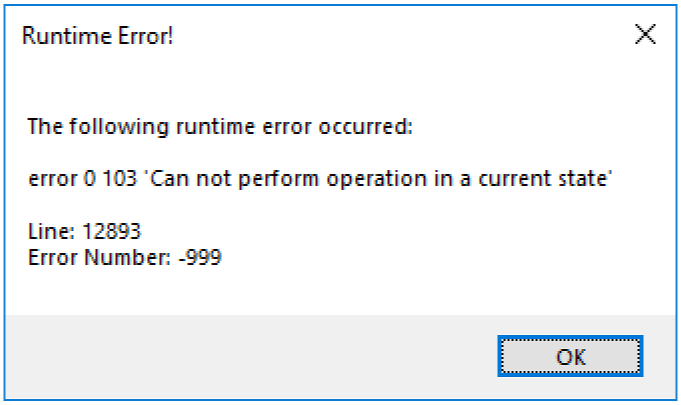


```
The following runtime error occurred:

Client Open call failed in Device Connect().  Host 'localhost (127.0.0.1)'
could not be reached on port 8080.  Specific endpoint failure: 'Project'.
Return code:'-102', Status code:'-102', Reason:''.  Please ensure Tobii
Pro Lab is Open and able to be reached with your current
TobiiProLabDevice settings.

Line: 11534
Error Number: 5502
```

The Project Type cannot be changed once it is created in a Project. Therefore, users need to create a new Project that uses the External Presenter Project Type to solve this issue.

Record Error

Before running a Tobii Pro Lab-enabled experiment, users need to navigate to the correct tab in Tobii Pro Lab. The Record tab must be selected before running a test session. The following error occurs if the Record tab is not selected:



```
The following runtime error occurred:

error 0 103 'Can not perform operation in a current state'

Line: 12893
Error Number: -999
```

For more information on configuring Tobii Pro Lab before running an experiment, please refer to Tutorial 5, Task 14: Prepare the Tobii Pro Lab Project.

Appendix A: Samples and Tutorials

The following table contains a list of all the Samples and Tutorials included with E-Prime Extensions for Tobii Pro. The folder the experiment is in and a short description are included.

Sample / Tutorial	Folder	Description
TETFixedPositionAOI	Samples/TET	TETFixedPositionAOI simultaneously displays a name and plays a sound of an animal. The experiment then displays two pictures of animals. The task is to identify which animal made the sound by choosing the corresponding picture of the animal. This experiment has TET gaze data file functionality implemented.
TETFixedPositionAOIMultiMonitor	Samples/TET	TETFixedPositionAOIMultiMonitor is a modified version of TETFixedPositionAOI that uses multiple monitors to present aspects of the experiment to the participant and the experimenter.
TETMovieRT	Samples/TET	TETMovieRT is a simple reaction time experiment that presents a fixation and a stimulus movie and collects a response to the stimulus. This experiment has TET gaze data file functionality implemented.
TETPictureRT	Samples/TET	TETPictureRT is a simple reaction time experiment that presents a Fixation and a stimulus picture and collects a response to the stimulus.
TETVaryingPositionAOI	Samples/TET	TETVaryingPositionAOI presents a letter in a physical orientation first, then shows many letters in several different orientations. The task is to find the orientation that matches the orientation of the first letter shown and press the one key once the letter with the same orientation is located.
TETVaryingPositionAOITracking	Samples/TET	TETVaryingPositionAOITracking is the same experiment as the non-tracking experiment. The main difference from TETVaryingPositionAOI is the tracking. As you fixate on a letter during the stimulus presentation, the BorderColor changes to show that the letter is currently being gazed at.
TPLFixedPositionAOI	Samples/TPL	TPLFixedPositionAOI is the same as TETFixedPositionAOI with the exception being it is configured for use with Tobii Pro Lab. Additionally, a gaze data file is not produced.
TPLMovieRT	Samples/TPL	TPLMovieRT is the same as TETMovieRT with the exception being it is configured for use with Tobii Pro Lab. Additionally, a gaze data file is not produced.
FixedPositionAOI	Tutorials/TET	FixedPositionAOI serves as the starting point for TET Tutorial (1). The experiment edited here is eventually used in Tutorials 2 and 3.
TETFixedPositionAOI	Tutorials/TPL	TETFixedPositionAOI serves as the starting point for the Tobii Pro Lab Tutorial. Basic TET-functionality is already added to save users time when modifying this Tutorial.

Appendix B: Hardware Configurations

The hardware configurations for the Tobii Pro eye tracker, E-Prime, and Tobii Pro computers, and their associated display screens are shown in the diagrams below. These diagrams assume DVI dual output graphic cards and DVI cables. If working with a DVI/HDMI/VGA card and/or utilizing HDMI or VGA connections, HDMI and VGA switches, splitters, and adapters can be used instead of or in addition to what is illustrated. Please refer to the Tobii Pro documentation for more hardware configurations and troubleshooting information.

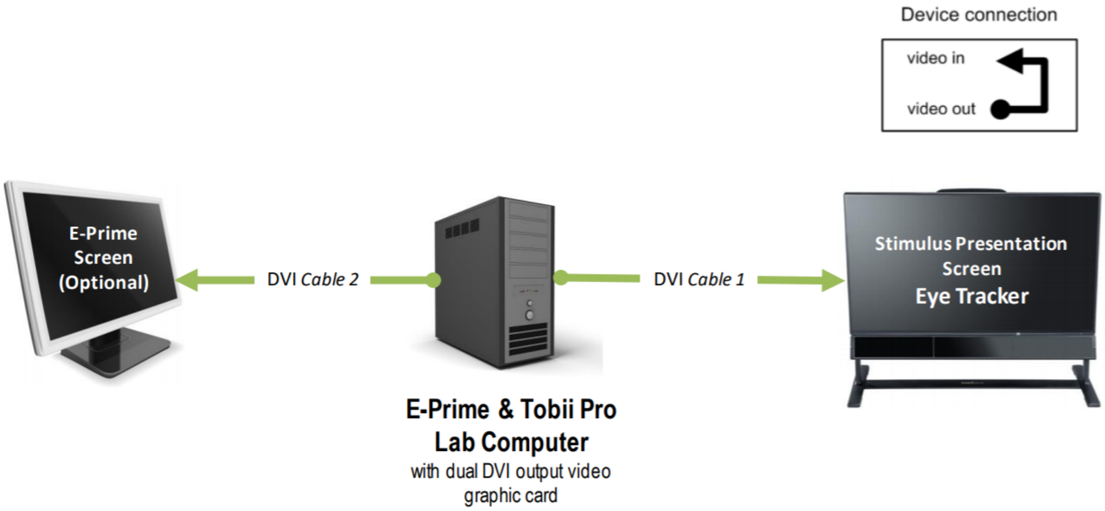
- The LAN cable used to connect the E-Prime Computer to other hardware devices can be connected directly to:
  - The Tobii Pro computer
  - A network or switch (do not use a hub) to which the Tobii Pro software and/or eye tracker computer is also connected
  - The USB adapter that connects to (and is delivered with) the eye tracker.

- The image from the stimulus presentation screen can be represented in the Tobii Pro software and should be shown on the eye tracker screen for the participant. Tobii Pro Lab can represent the stimulus display events by using the PackageCall TPLSetDisplayEvent (see *Tutorial 5: Adding Tobii Pro Lab Support to an Experiment*). It records information about the display events sent from E-Prime to recreate the designated screens displayed during the experiment. Tobii Pro Studio requires a video capture card to record the stimulus presentation object.

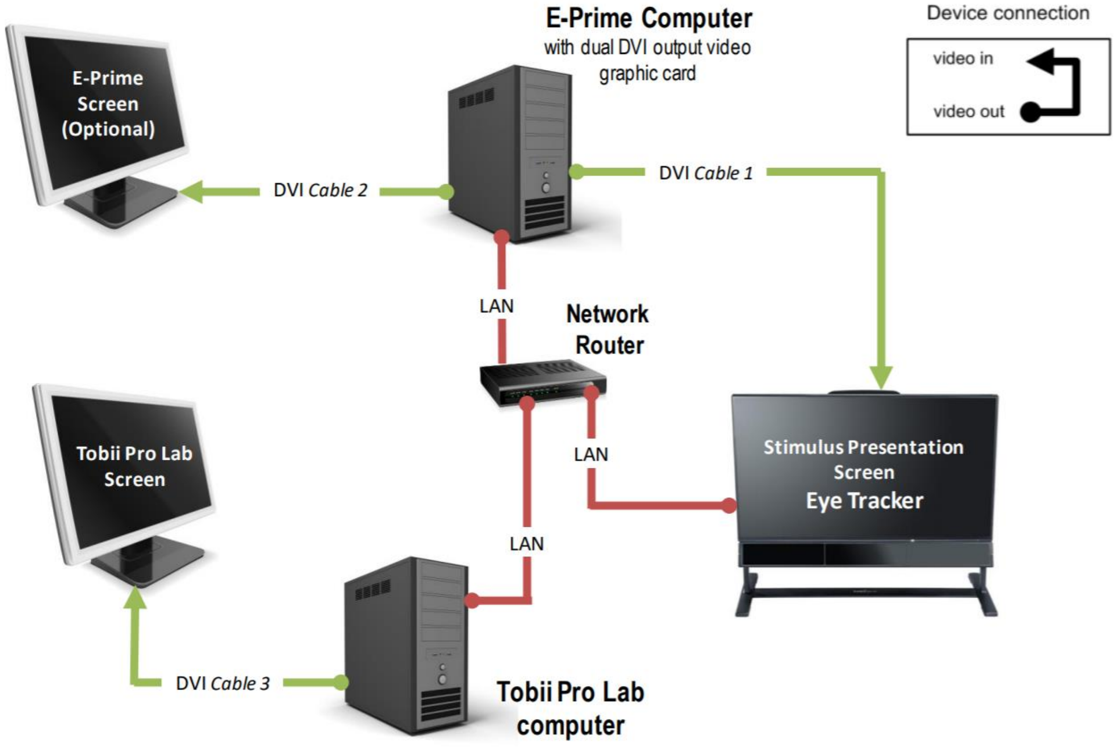
- In *Hardware Configuration with Two Machines (Calibration in Tobii Pro Lab)*, to switch between input from the E-Prime computer and the Tobii Pro computer, press the DVI switch (if equipped).

- To show the image from the E-Prime computer on a separate experimenter screen, use a splitter or dual graphics card with the E-Prime computer and the Stimulus Presentation screen with the eye tracker (see *Tutorial 3: Multiple Monitors: Creating a Participant Station*).

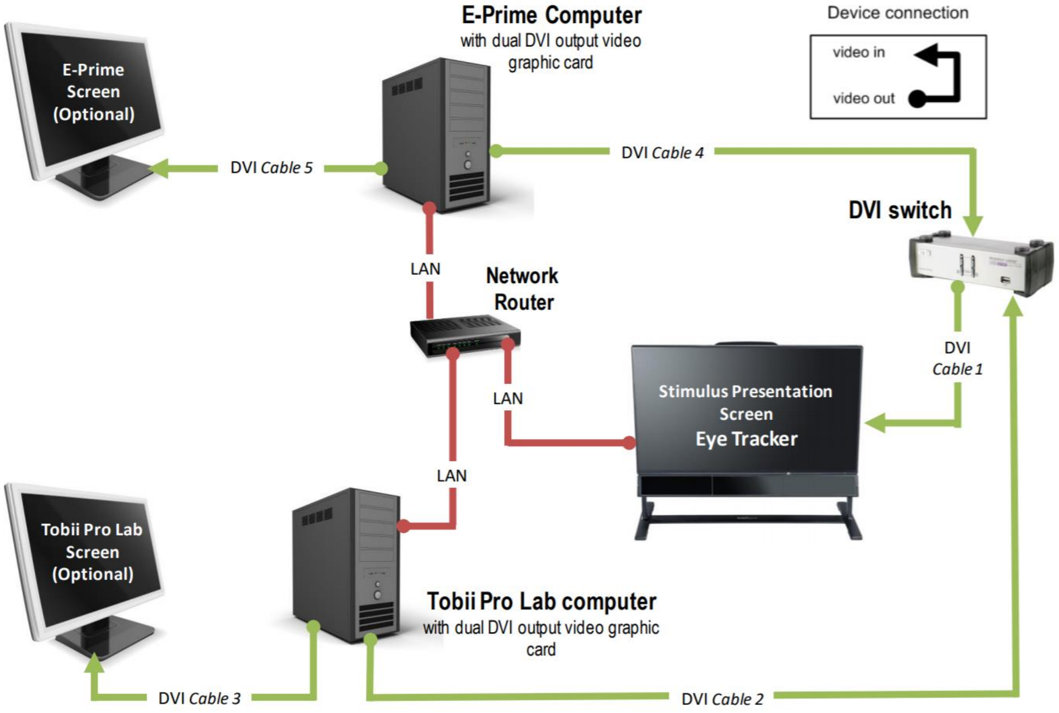
B.1 Hardware Configuration with One Machine


Device connection
video in
video out
E-Prime Screen (Optional)
DVI Cable 2
DVI Cable 1
Stimulus Presentation Screen
Eye Tracker
E-Prime & Tobii Pro Lab Computer
with dual DVI output video graphic card

B.2 Hardware Configuration with Two Machines (Calibration in E-Prime – Preferred)



B.3 Hardware Configuration with Two Machines (Calibration in Tobii Pro Lab)



Appendix C: Timing and Synchronization: Writing to the Buffer and Buffer Size

It is important to make sure all the data from the experiment is being written to the gaze data file. To do this, we recommend using the gaze data PackageCalls and modify them for the experiment (refer to *Tutorial 2: Logging Gaze Data in an Experiment*). There are two important pieces of information to note when collecting data: buffer size and when the data is written to the buffer.

The buffer starts to fill when TETStartTracking is called and stops when TETStopTracking is called. Contained in TETStartTracking PackageCall is an option to clear the buffer when the call is made (i.e., the parameter vClearHistory which is True by default). This option ensures the buffer is empty and ready to be filled with data every time TETStartTracking is called. Additionally, the TETClearHistory PackageCall is available to manually clear the buffer at other times during an experiment. TETGazeDataSave clears the buffer by default at the end of each trial. However, this can be changed with the TETGazeDataSave parameters (see *Task 4: Add the TETGazeDataSave PackageCall to Save Eye Tracking Data*).

Gaze data should be written during a non-critical timing moment such as at the end of a trial. The aim of this is to prevent disruption of the experiment timing. Therefore, the buffer size selected in the experiment should cover all the samples collected during a trial. Contained in the table below are eye tracker Frequencies (Hz), Max Count (Samples), and the corresponding number of seconds to fill the buffer at a given speed. The default Max Count is 100,000 but should be changed if it is not enough for the experiment. The maximum size buffer you can create may be limited by the memory of your computer.

		Time to Fill Buffer (secs)					
		Eye Tracker Speed (Hertz)					
		30Hz	**60Hz**	**120Hz**	**300Hz**	**600Hz**	**1200Hz**
Max Count (Samples)	**1000**	33.333	16.667	8.333	3.333	1.667	0.833
	2000	66.667	33.333	16.667	6.667	3.333	1.667
	4000	133.333	66.667	33.333	13.333	6.667	3.333
	5000	166.667	83.333	41.667	16.667	8.333	4.167
	10000	333.333	166.667	83.333	33.333	16.667	8.333
	50000	1666.667	833.333	416.667	166.667	83.333	41.667
	100000	**3333.333**	**1666.667**	**833.333**	**333.333**	**166.667**	**83.333**
	250000	**8333.333**	**4166.667**	**2083.333**	**833.333**	**416.667**	**208.333**
	500000	**16666.667**	**8333.333**	**4166.667**	**1666.667**	**833.333**	**416.667**
	1000000	**33333.333**	**16666.667**	**8333.333**	**3333.333**	**1666.667**	**833.333**

5. Contact Information

PSYCHOLOGY
SOFTWARE TOOLS
Solutions for Research, Assessment, and Education

For additional information or support contact us at:

Psychology Software Tools, Inc.
311 23rd Street Extension, Suite 200
Sharpsburg, PA 15215-2821
Phone: 412-449-0078
Fax: 412-449-0079
www.pstnet.com

For Product Service and Support:
https://support.pstnet.com