

Psychtoolbox for Matlab and Octave comes with native support for the EyeLink eye tracking systems and contains a fully featured wrapper for the EyeLink C API via the EyeLink Toolbox. This wrapper requires the EyeLink Developers Kit be installed on your machine (Windows, macOS, or Linux) to operate properly, but is otherwise completely integrated within your Psychtoolbox installation. Psychtoolbox and the EyeLink Developers Kit are license free.

NOTE: While the EyeLink Developers Kit will work for Windows, macOS, and Linux, Psychtoolbox has it own [system requirements and recommendations](#)

Installation Instructions:

1. [Configure your Display PC's network connection to communicate with the EyeLink Host PC.](#)
2. Install the latest version of [Psychtoolbox](#) within Matlab or Octave.
3. Install the latest version of the [EyeLink Developers Kit](#) for your operating system.

The EyeLink Developers Kit installs the EyelinkToolbox and the necessary mex files in the following location:

- Windows: C:\Program Files (x86)\SR Research\EyeLink\SampleExperiments\MATLAB Psychtoolbox
- macOS: /Applications/Eyelink/SampleExperiments/MATLAB Psychtoolbox
- Linux: /usr/share/EyeLink/SampleExperiments/MATLAB Psychtoolbox

4. Run the **SetupSREyeLinkToolbox.m** script that can be found in the folders listed above. This script will backup and replace the EyeLink Toolbox that is packaged with Psychtoolbox.

Testing your installation

You can find a list of all the available functions in the EyeLink toolbox by typing "Eyelink" into the Matlab terminal followed by pressing Enter. To check that everything is working, please run one of the example scripts described below.

*****Please note that recent versions of psychtoolbox require InitializePsychSound to be run before running our example scripts*****

EyeLink Toolbox Templates / Example Projects:

The EyeLink Toolbox contains a number of example scripts which illustrate how to properly integrate EyeLink Toolbox functions into Psychtoolbox projects. You can find these examples in the following directory after running the **SetupSREyeLinkToolbox.m** script:

<Psychtoolbox installation directory> -> PsychHardware -> EyelinkToolbox -> EyelinkDemos -> SR-ResearchDemos

NOTE: There are other user-created demos in the EyeLinkDemos folder but we highly recommend sticking with the examples in the *SR-ResearchDemos* folder, as the other demos may utilize deprecated functions and are not designed to be compatible with Data Viewer's integration protocols.

The following example scripts are provided:

Templates	Description
<i>SimplePicture</i>	A simple EyeLink integration demo that records eye movements passively while an image is presented on the screen. Each trial ends when the space bar

	or a button is pressed.
<i>StereoPicture</i>	EyeLink integration demo for stereo presentation. Records eye movements passively while presenting a stereo stimulus. Supports both split-screen mode and dual-monitor setup. Each trial ends when the space bar is pressed. Data Viewer integration with both left and right eyes superimposed on the same eye window view.
<i>PursuitTarget</i>	A smooth pursuit EyeLink integration demo that records eye movements while a target moves sinusoidally across the screen. Each trial ends after 5s.
<i>SimpleVideo</i>	Simple video demo with EyeLink integration and animated calibration / drift-check/correction targets. In each trial eye movements are recorded while a video stimulus is presented on the screen. Each trial ends when the space bar is pressed or the video stops playing. A different drift-check/correction animated target is used in each of the 2 trials.
<i>MRI_BlockRecord</i>	Simple MRI demo with EyeLink integration. 6 trials are presented in 2 blocks of 3 trials. Trial duration is 5.5s during which a 4s stimulus is presented. A block starts with a drift-check followed by presentation of central crosshairs. Eye movements are recorded while waiting for an MRI trigger (keyboard key 't' in this demo). The stimulus is presented when trigger is received. A fixed ITI is maintained by presenting crosshairs between each 4s stimulus. Eye movements are recorded throughout an entire block rather than on a trial-by-trial basis.
<i>GazeContingent -> FixWindowBufferedSamples</i>	EyeLink gaze-contingent demo that shows how to retrieve online gaze samples from a buffer. In each trial central crosshairs are shown until gaze is detected continuously within a central square window for 500ms or until the space bar is pressed. An image is then presented until the space bar is pressed to end the trial.
<i>GazeContingent -> FixWindowFastSamples</i>	EyeLink gaze-contingent demo showing how to retrieve fast gaze samples online. In each trial central crosshairs are shown until gaze is detected continuously within a central square window for 500ms or until the space bar is pressed. An image is then presented until the space bar is pressed to end the trial.
<i>GazeContingent -> GCBufferedEvents -> BufferedEndSacEvents</i>	A simple EyeLink gaze-contingent demo showing how to retrieve online events from a buffer. In each trial an image is presented with a red gaze-contingent dot overlaid on top. The dot's location is updated online based on the end x, y coordinates of each saccade detected. Each trial ends when the space bar is pressed.
<i>GazeContingent -> GCBufferedEvents -> BufferedFixUpdateEvents</i>	A simple EyeLink gaze-contingent demo showing how to retrieve online events from a buffer. In each trial an image is presented with a red gaze-contingent dot overlaid on top. The dot's location is based on the average x, y coordinates of fixations updated online every 50ms via a FIXUPDATE event.
<i>GazeContingent -> GCFastSamples</i>	A simple EyeLink gaze-contingent demo showing how to retrieve fast online samples. In each trial an image is presented with a red gaze-contingent dot overlaid on top. The dot's location is updated online based on the x, y coordinates of the latest gaze sample retrieved online. Each trial ends when the space bar is pressed.

EyeLink Data Viewer Messaging Commands and Syntax:

Throughout the course of your experiment, in addition to sending messages to the EyeLink data file (i.e., EDF

file) that log the timing of events within your task, it's typically a good idea to send special messages that allow integration with Data Viewer. The examples included in the EyeLink Developers Kit illustrate how to send special messages to the EDF to convey important information to Data Viewer, like trial starting/ending times, trial variable values, trial stimulus information, and more. You can find a webinar at the following link that covers the formatting of these messages in detail:

- [EyeLink Data Viewer Integration Messaging Protocol](#)

You can also find full documentation of the protocol in the [Data Viewer User Manual](#) (Help -> Contents) under the section "Protocol for EyeLink Data to Viewer Integration".

Documentation:

This video tutorial series provides a brief overview of the basics of integrating EyeLink systems via Matlab/Psychtoolbox and the EyeLink Toolbox:

- [EyeLink Psychtoolbox Video Tutorial Series](#)

The following Webinar also contains lots of useful information:

- [Programming EyeLink Experiments Using Psychtoolbox for Matlab](#)

The EyeLink Toolbox is a wrapper for our API contained within the [EyeLink Developers Kit](#), and the [EyeLink Programmer's Guide](#) contains the complete documentation of our C API functions, data structure, and implementation. The syntax functions listed in the manual will likely differ in how they are wrapped within the EyeLink Toolbox, but much of the functionality should be analogous.