



Startup Guide

Matlab

v 1.1.9

September 2023



Introduction

About

The EyeLogic Software Development Kit (SDK) is a free software package for building custom applications which use an EyeLogic eye tracking device. It offers the possibility to connect with your device via an application programming interface (API) from any custom application. The EyeLogic SDK is available for the programming languages

- C++,
- C#,
- C,
- Python, and
- Matlab

It is also usable with any other programming language that is capable of importing dynamic link libraries (DLLs), e.g. Visual Basic.

For each directly supported language, there is a short and simple sample program to help you get started with the development of your first EyeLogic application.

This guide describes the use of the EyeLogic API for Matlab and gives a step-by-step introduction on how to start with your own Matlab program.

System Requirements

For the system requirements of the EyeLogic Server and an installation guide, please refer to the Server's documentation.

The SDK has no additional requirements. It is built for Microsoft Windows only (32 bit or 64 bit).

Installation and Getting Started

Download Software

In order to use an EyeLogic eye tracking device from within your application, you need the EyeLogic Server and the EyeLogic SDK. Check the download-page to get the latest release of

both packages: <https://www.eyelogicsolutions.com/help-center/>

Compatibility

The software is written to support backwards-compatibility, i.e. an update of the EyeLogic Server software will not break support for your device, irregardless of the model. The actual guide assumes that you are installing the newest version of the EyeLogic Server. Please always update to the newest server version before reporting an error to the EyeLogic support.

On the other hand, updating the SDK and API-DLLs is not always necessary. Since you as a programmer would have to recompile your application with every SDK-update, we designed the SDK such that the server is able to communicate with older API versions. Therefore, when shipping your application, just add the EyeLogic API DLLs of the actual version to your package. It is compatible with servers of the actual and newer releases.

See Section Shipping your Application for a tutorial on how to ship your application.

Install EyeLogic SDK on Windows

The EyeLogic SDK does not need to be installed. It ships as .zip file which just needs to be extracted to some directory on your hard disk. Be sure, that you have user-rights to that directory, e.g. any directory inside C:\Program Files or similar is problematic, since it requires admin rights to access those files every time you start your program. It is recommended to use a user-local directory.

Note: The SDK has to be installed on the same computer as the server. Please see the main server manual for help on installing the server.

After extracting the .zip file, the directory contains one subfolder for each supported programming language. Open the matlab folder, the content should be:

1. eyelogic - contains the Matlab package which binds to the EyeLogic API
2. demo_basicusage - contains a sample Matlab script which demonstrates the use of the EyeLogic API

Getting Started with the Sample Code

In the directory, into which you unpacked the SDK **EyeLogicSDK**, navigate to the sub-directory **matlab**. Open the demo_basicusage\Sample.m file with Matlab.

You may run the application by pressing F5 or by clicking the "Run" button. The location of the script needs to be either the current directory or listed in the Matlab paths. On the first run,

Matlab asks if you want to change the current directory or add the script location to the path list. Choose whatever is more suitable to you, both choices work.

Before running the application check that the EyeLogic Server is running (see the EyeLogic Server manual). If the server is running, there is an EyeLogic icon in the windows tray bar.

Note that your firewall might block the connection between your program and the server. In this case, add a rule to your firewall to allow your application to open TCP/UDP ports to an application on localhost (for the windows defender, just click "accept").

If you reached this point, you have properly set up your EyeLogic SDK. You may now start with the development of you own application. See the next section for the basic programming concepts and for a tutorial on how to deploy and ship your application.

Concepts

Server-Client Setup

The EyeLogic software consists of two main parts: The **server** and the **API**. The server is the neccessary driver for your eye tracking device. It detects your device and handles the communication. The API is part of the EyeLogic Standard Development Kit (SDK). It consists of .dll files which can be used by your application to set up a connection to the EyeLogic Server, start tracking and receive eye tracking data.

The server is designed to run permanently on your computer as a background process. While not actively tracking the server requires an insignificant portion of your computer's resources. Once an EyeLogic eye tracking device is plugged in, the server application detects it automatically and allows the user to set it up via the servers' configuration dialog (see the server icon in the windows tray bar). If for any reason the server background process is not running (the tray icon is missing), you may start the server manually via the windows start menu.

The API is a set of .dll files which can be used by any custom program (called the **user application**). With these DLLs the user application can establish a connection to the (running) server via the TCP protocol.

Set Up a Project for your Application

For an easy start to develop a new application it is recommended to copy the existing sample folder to a new location (e.g. EyeLogic_SDK\matlab with all its contents). The sample source file already provides a fully functional implementation. Starting from this sample code, you can easily modify and extend the code to suit your customized experiment.

Alternatively you can start a new Matlab project from scratch. In that case be sure that your

development environment is able to find the path for the EyeLogic matlab module (which is **<Location of your EyeLogic_SDK>\matlab\eyelogic**).

Control Flow between API and server

The usual control flow between the custom application/API and server is characterized by the following steps:

- **initialize:** Before calling any other function the API DLLs need initializing.
- **connect to server:** Establish a connection to the server via TCP.
- **find eye tracking device:** Obtain an info of connected eye trackers, otherwise wait until an eye tracker is plugged in.
- **start tracking:** Request tracking. If successful, the device will start tracking and the server reports **GazeSamples** to the user application, see also Section GazeSamples.
- **perform calibration:** Request a calibration. The screen will show some calibration points which the user has to look at. After calibration the system is calibrated and ready to use.
- **shut down:** At the end of your experiment either stop the tracking or simply shutdown the API.

All information which is passed from the server to the user application will be passed via **asynchronous callbacks**. The application has to register those callback functions in the API (see Section Example Program for an example implementation).

Note that you need to calibrate in order to obtain valid gaze samples (see Section GazeSamples). All gaze samples which are reported before the system is calibrated contain no valid eye data.

Example Program

In this section, the code of the Matlab example program is explained in some detail.

The file starts with importing the EyeLogic class.

```
api = EyeLogicAPI;
```

Note that the folder **<SDK>\matlab\eyelogic** must be added to the Matlab paths before, using the method **addpath()**.

In the next section, the application implements its control flow. It consists of the following code lines:

```
retConnect = api.connect();
```

Connects to the EyeLogic server. Check for the return value in order to find out whether the connection was established successfully.

```
screenConfig = api.getActiveScreen();
```

and

```
device = api.getDeviceConfig();
```

are called in order to obtain information about the active screen and the connected eye tracking device. If no device is connected, then the returned device is 'empty' (check by calling `isempty(device)`).

```
retStart = api.requestTracking(0);
```

Tells the device to start tracking and the Server to begin sample processing. The parameter 0 specifies the frame rate mode. If your device is capable of multiple frame rate modes (60Hz, 120Hz or 250Hz), you can also enter a different number. The list of available frame rate modes is stored in the member **frameRates** of **device**, whereas the first frame rate mode (0) is the default mode, which usually is the highest available speed mode of your system.

```
retCalibrate = api.calibrate(0);
```

Performs a calibration. This method blocks until the calibration ends - i.e. completed or aborted. The parameter 0 denotes the type of calibration. A list of available calibration methods is stored in member **calibrationMethods** of **device**.

The example program waits for 5 seconds and collects incoming GazeSamples. Since the Matlab SDK does not allow for callback functions, the only way of obtaining GazeSamples is to regularly call

```
api.getNextGazeSample( ms )
```

in a loop. If you specify a timeout **ms** in milliseconds, the method blocks until a GazeSample arrives or the timeout is reached. Check the return value in order to find out if there was a new GazeSample in that time. You may also pass `ms=0` if you do not want that the method blocks, but just want to check for and collect a new incoming GazeSamples.

Note, that beside GazeSamples, the EyeLogic API may send you Events which are notifications e.g. for a disconnected eye tracking device. In order to obtain all events and GazeSamples, you have to call both

```
api.getNextGazeSample( ms )
```

and

```
api.getNextEvent( ms )
```

alternating. Note that they may be more than one event/GazeSample pending at a time. A call to one of the methods obtains just the next event/GazeSample in the buffer.

Finally, the connection to the EyeLogic Server is closed.

```
api.unrequestTracking()  
api.disconnect()  
api.delete()
```

GazeSamples

GazeSamples are the most essential data which is generated by the eye tracker. The eye tracker delivers one GazeSample per frame. Each sample contains information on the time of measurement, the position of the eyes, the pupil radius and the point where the user looks at on some stimulus plane (usually a computer monitor).

Shipping your Application

When you want to ship your application, be sure to include all relevant files so that it may run on different computers. The EyeLogic functionality will only work on computers which have the EyeLogic Server installed. The installed server needs to at least be of the same version as the shipped API DLLs (a newer server version is permissible).

Beside the relevant files of your application, you need to ship the eyelogic/ subfolder (from **SDK path/matlab** with all its content. The matlab interpreter must be able to locate this folder in order to locate the eyelogic module on your destination machine. You may place the eyelogic/ folder inside the working directory of your application and ship them altogether.

Appendix

License Agreement and Warranty for SDK

IMPORTANT - PLEASE READ CAREFULLY:

The License Agreement is a legal agreement between you and EyeLogic GmbH and its affiliates ("EyeLogic", "we", or "us"). This license agreement governs your use of the EyeLogic software and any third party software that may be distributed therewith (collectively the "software"). EyeLogic agrees to license the software to you (personally and/or on behalf of your employer) (collectively "you" or "your") only if you accept all the terms contained in this license agreement. By installing, using, copying, or distributing all or any portion of the software, you accept and agree to be bound by all of the terms and conditions of this license agreement.

If you do not agree with any of the terms of this license agreement, do not install or use the software.

1. **License Grant:** EyeLogic grants you a revocable, nonexclusive, non-transferable, limited right to install and use the application on a device owned and controlled by you, and to access and use the application on such mobile device strictly in accordance with the terms and conditions of this licenses, the usage rules and any service agreement associated with your device. The Software includes third party software and other copyrighted material. Acknowledgements, licensing terms and disclaimers for such Third Party Software are provided with the Software or contained in the Documentation, and your use of such Third Party Software is governed by their respective terms (collectively "Related Agreements").
2. **Restriction on Use:** You shall use the application strictly in accordance with the terms of the related agreements and shall not:
 - (a) decompile, reverse engineer, disassemble, attempt to derive the source code of, or decrypt the application,
 - (b) make any modification, adaption, improvement, enhancement, translation or derivative work from the application,
 - (c) violate any applicable laws, rules or regulations in connection with your access or use of the application,
 - (d) remove, alter or obscure any proprietary notice (including any notice of copyright or trademark) of EyeLogic or its affiliates, partners, suppliers or the licensors of the application,
 - (e) use the application for any revenue generating endeavor, commercial enterprise or other purpose for which it is not designed or intended,
 - (f) make the application publicly available over a network or other environment permitting access or use by others without the written permission of EyeLogic,
 - (g) use the application for creating a product, service or software that is, directly or indirectly, competitive with or in any way substitute for any services, product or software offered by EyeLogic,
 - (h) use any proprietary information or interfaces of EyeLogic or other intellectual property of EyeLogic in the design, development, manufacture, licensing or distribution of any applications, accessories or devices for use with the application.

3. **Termination:** EyeLogic may, in its sole and absolute discretion, at any time and for any or no reason, suspend or terminate this license and the rights afforded to you hereunder with or without prior notice. Furthermore, if you fail to comply with any terms and conditions of this license, then this license and any rights afforded to you hereunder shall terminate automatically, without any notice or other action by EyeLogic. Upon the termination of this license, you shall cease all use of the application and uninstall the application.
4. **Disclaimer of Warranties:** You acknowledge and agree that the application is provided on an “as is” and “as available” basis, and that your use of or reliance upon the application and any third party content and services accessed thereby is at your sole risk and discretion. EyeLogic and its affiliates, partners, suppliers and licensors hereby disclaim any and all representations, warranties and guarantees regarding the application and third party content and services, whether express, implied or statutory, and including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, furthermore, EyeLogic and its affiliates, partners, suppliers and licensors make no warranty that
- (a) The application or third party content and services will meet your requirements
 - (b) The application or third party content and services will be uninterrupted, accurate, reliable, timely, secure or error-free
 - (c) The quality of any products, services, information or other material accessed or obtained by you through the application will be as represented or meet your expectations, or
 - (d) Any errors in the application or third party content and services will be corrected.

No advice or information, whether oral or written, obtained by you from EyeLogic or from the application will create any warranty not expressly made herein or create any liability on the part of EyeLogic.

If the licensee modifies or replaces any of the third party open source software included in the software, EyeLogic is not obligated to provide any updates, maintenance, warranty, technical or other support or services for the resultant modified Software. You expressly acknowledge that any failure or damage to any hardware, software or systems as a result of such modification to the open source components of the software is excluded from the terms of any EyeLogic warranty.

5. **Limitation of liability:** Under no circumstances shall EyeLogic or its affiliates, partners, suppliers or licensors be liable for any indirect, incidental, consequential, special or exemplary damages arising out of or in connection with your access or use of or inability to access or use the application and any third party content and services, whether or not the damages were foreseeable and whether or not EyeLogic was advised of the possibility of such damages. Without limiting the generality of the foregoing, EyeLogic’s aggregate liability to you (whether under contract, tort, statute or otherwise) shall not exceed the amounts actually paid by licensee for the licensed materials. The foregoing limitations will apply even if the above stated remedy fails of its essential purpose.

6. **Confidentiality:** Licensed materials are proprietary to EyeLogic and constitute EyeLogic trade and business secrets. The licensee shall maintain licensed materials in confidence and prevent their disclosure using at least the same degree of care it uses for its own trade and business secrets, but in no event less than a reasonable degree of care. The licensee shall not disclose licensed materials or any part thereof to anyone for any purpose, other than to its employees and sub-contractors, if any, for the purpose of exercising the rights expressly granted under this agreement, provided they have in writing agreed to confidentiality obligations at least equivalent to the obligations stated herein. The foregoing does not apply to information that

- (a) is or becomes generally known or available to the public without any breach of the confidentiality obligation by licensee
- (b) was already known to licensee prior to the disclosure by EyeLogic, or
- (c) was rightfully acquired by licensee from a third party without a breach of a confidentiality obligation towards EyeLogic.

In case of a dispute, the licensee has the burden of proof that the licensed materials and/or any portion thereof fall under one of these exceptions. Should the licensee be legally compelled to disclose any licensed materials to a third party, such as pursuant to a mandatory order by a court or authority or any comparable action, the licensee shall, to the extent permitted under applicable law, inform EyeLogic without undue delay and undertake all possible measures to safeguard secrecy.

About EyeLogic

EyeLogic is a manufacturer of high precision and high quality eye tracking devices, mainly for scientific and research use cases. EyeLogic GmbH is a spin-off of the Free University of Berlin, faculty of mathematics and computer science and has a vast experience in image processing and computer vision.

Contact and Support

For technical support questions contact us via mail at: `<code>support@eyelogicsolutions.com</code>`

EyeLogic GmbH
Schlesische Straße 28
D-10997 Berlin
Germany

Phone: +49 30 9841 9241
Mail: info@eyelogicsolutions.com
Web: <https://www.eyelogicsolutions.com>

Copyright (C) 2019-2023 EyeLogic GmbH